INTERFACE AGE™

MICROCOMPUTING FOR CONSUMER AND BUSINESS APPLICATIONS VOLUME 3, ISSUE 5 MAY 1978 $2.00
CANADA/MEXICO $2.50 INTERNATIONAL $3.50

Inside ASCII – Part I

Floppy ROM™ #4
in the IAPS™ Format

Author Index – 1975-1977

VOL. 3, ISSUE 5
SINCE DECEMBER 1975

MAY 1978

COVER STORY

This month's cover, supplied by Space Byte, represents the emergence of the microcomputer from the realm of an idea to a useful business tool.

The cover captures the growth of the microcomputer industry and represents the freedom it provides to the business world. With the use of microcomputer systems, the traditional methods of bookkeeping and paper world can be tossed away to disappear into the oblivion of today's and tomorrow's technology.

INTERFACE AGE

MICROCOMPUTING FOR CONSUMER AND BUSINESS APPLICATIONS

BUSINESS FEATURES

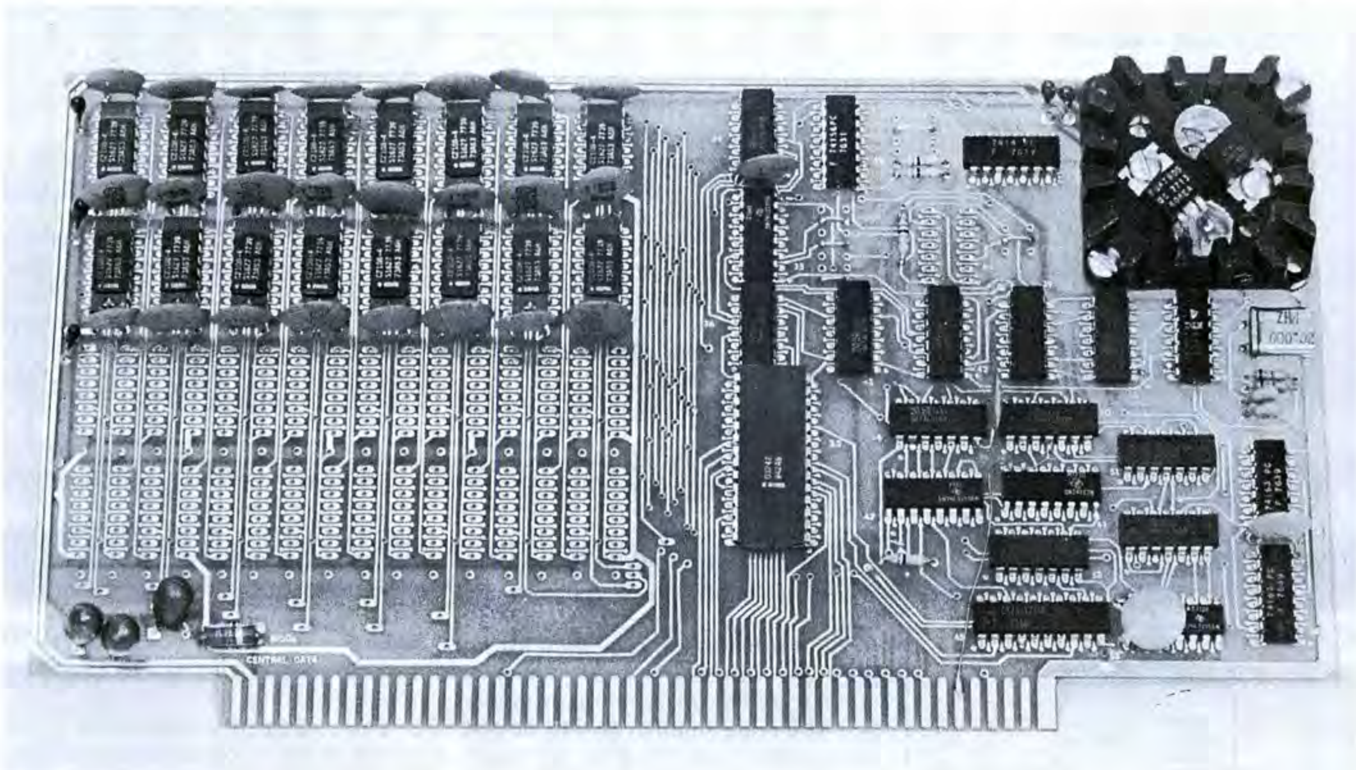
GENERAL FEATURES

HARDWARE FEATURES

SOFTWARE FEATURES

INTERFACE AGE Magazine, published monthly by McPheters, Wolfe & Jones, 16704 Marquardt Ave., Cerritos, CA 90701. Subscription rates: U.S. $14.00, Canada/Mexico $16.00, all other countries $24.00. Make checks payable in U.S. funds drawn on a U.S. bank. Opinions expressed in by-lined articles do not necessarily reflect the opinion of this magazine or the publisher. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsement of the product or products by this magazine or the publisher.
 INTERFACE AGE Magazine is catalogued in the Library of Congress, Classification No. QA75.5.155. Membership in Audit Bureau of Circulations applied for.
POSTMASTER: Please send change of address form 3579 and undelivered copies to **INTERFACE AGE** Magazine, 16704 Marquardt Ave., Cerritos, CA 90701. Second-class postage paid at Artesia, California 90701 and at additional mailing offices.

Memory Makes the Computer, Central Data Makes the Memory



Memory is an important part of a computer system. So don't even think of settling for a poor quality RAM board. Central Data offers carefully engineered 16K, 24K and 32K RAM boards to suit your needs. We take our RAM boards seriously, so should you.

Reliability. Each one of our boards comes completely assembled, tested and burned in. No fiddling with an infinite number of parts from a kit or costly investment in equipment to test your board. Each one of our boards is backed by a one year warranty to insure your satisfaction.

Expandability. We offer a 16K RAM board with capacity to 32K. You can purchase a 16K board and expand later or buy a full 32K.

Performance. Unlike some dynamic memory boards, ours uses an invisible refresh that inserts refresh cycles between your computer's normal memory access cycles. This eliminates loss in performance of your computer with lengthy refresh cycles.

Power. Our board uses less than one-half the power of a static RAM.

Price. You can buy a 16K RAM board, completely ready to use, for only $289. A full 32K RAM board is available for $475, or you can add on later for only $200.

When you buy our memory board, you'll be buying the best equipment at the best price. Order yours today by mail or phone.

Please send me the following:
- ☐ 16K RAM board ($289)
- ☐ 32K RAM board ($475)
- ☐ additional information

I enclose $________.
Ill. residents add sales tax.
Central Data P.O. Box 2484, Sta. A Champaign, Il 61820

Name ________
Address ________
City ________
State ________
Zip ________

Central data

P.O. Box 2484, Sta. A
Champaign, Il 61820
(217) 359-8010

INTERFACE AGE™

16704 Marquardt Avenue
Cerritos, CA 90701
(213) 926-9544

PUBLISHER & EDITOR-IN-CHIEF
ROBERT S. JONES

EXECUTIVE PUBLISHER
NANCY A. JONES

ASSISTANT-TO-THE-PUBLISHER
KATHY ROBERTSON

GENERAL MANAGER
EVA YAKA

MARKETING MANAGER
MEL MURCHISON

MARKETING ASSISTANT
DENISE JACKSON

ADMINISTRATION
PUBLICATION DIRECTOR
MIKE ANTICH

CIRCULATION MANAGER
DONNA YOSHIOKA

SUBSCRIPTION CIRCULATION
JO ANN FERGUSON

DOMESTIC RETAIL CIRCULATION
ZACH BOVINETTE
(213) 795-7002

ACCOUNTING
SAYOKO TANISAKI

EDITORIAL
SENIOR EDITOR
CARL WARREN

ASSISTANT EDITOR
SANDI MC KEEN

NORTHWESTERN REGIONAL EDITOR
ADAM OSBORNE, PhD

NORTHEASTERN REGIONAL EDITOR
ROGER C. GARRETT

SOUTHEASTERN REGIONAL EDITOR
WILLIAM TURNER

SOFTWARE EDITOR
ABRAHAM A. PEREZ, PhD

HARDWARE EDITOR
ROGER EDELSON

CONTRIBUTING BUSINESS EDITOR
BUD SHAMBURGER

CONTRIBUTING EDITOR
ALAN R. MILLER, PhD

Editorial Correspondence
Direct all correspondence to the appropriate editor at: INTERFACE AGE Magazine, P.O. Box 1234, Cerritos, CA 90701

PRODUCTION
ART
MARGARET FENSTERMAKER
FINO ORTIZ

PRODUCTION ARTISTS
SHELLEY WRIGHT
SANDI EVANS

TYPOGRAPHER
MELODY A. MARTENS

ADVERTISING
NATIONAL SALES MANAGER
BRUCE BERKEY

ADVERTISING-Representatives
TIM O'ROURKE-ZACH BOVINETTE

Advertising Inquiries
Direct all advertising inquiries to: Advertising Dept., INTERFACE AGE Magazine, 61 South Lake Avenue, P.O. Box 4566, Pasadena, CA 91106 (213) 795-7002.

FOREIGN CIRCULATION
JAPAN CIRCULATION
KAZUHIKO NISHI
ASCII Publishing - 305 HI TORIO
5-6-4 Minami Aoyama,
Minato-ku, Tokyo 107 Japan
Telephone: (03) 407-4910

UNITED KINGDOM CIRCULATION
VINCENT COEN
L.P. Enterprise, 313 Kingston Road,
Ilford, Essex, England IGIIPJ.
Telephone: 01-553-1001

WESTERN CANADA CIRCULATION
BRIAN I.J. WIEBE
Kitronic, 25236 26th Avenue RR5
Aldergrove, B.C. VOX 1A0
Telephone: (604) 856-2301

EASTERN CANADA CIRCULATION
LIZ JANEK
RS-232, Ltd., 186 Queen Street West,
Toronto, Ontario M5V 1Z1
Telephone: (426) 598-0288

AUSTRALIA CIRCULATION
R. J. HOESS
Electronic Concepts Pty. Ltd.
52-58 Clarence Street, Sydney NSW 2000
Telephone: 29-3753

MEMBER OF THE WESTERN PUBLISHERS ASSOCIATION

AMERICAN SOCIETY OF BUSINESS PRESS EDITORS

EDITOR'S

By the time you read this column, several major trade shows will have taken place. The West Coast Computer Faire, followed by Micro Business '78 in Pasadena, both held in March, then PERCOM '78 held in April.

The West Coast Computer Faire was as usual exciting, dynamic and worthwhile. Micro Business '78, a brand new show sponsored by INTERFACE AGE, was the first of its kind in that it was geared completely to the small business user.

Micro Business '78, although small when compared to the Computer Faire, was a total success. For the first time, prime manufacturers of small business systems were able to get together and talk to an interested group of potential users.

All three shows had as their primary goal to show what is out there in computer land, and provide a forum for both the industry and the user to come together on common grounds.

If you were unable to visit any of these shows, don't give up hope; there are plenty more on the horizon, such as NCC in June and Personal Computing in August. For those of you who are new to computing, or are interested in keeping up on what's happening, it is important to attend at least one show a year. Believe me, you will find at least a dozen booths, at any given show, to hold your interest.

In line with the great big shows is a little show, or more correctly a swap meet, being put together by a colleague, John Craig. John is holding the Southern California Swap Meet, July 1, 1978 at the Santa Barbara Salvation Army Community Center. For more information on John's get together write to: John Craig, RFD, Box 100D, Lompoc, CA 93436.

Along with trade shows, rolling in by the dozens, are new publications, and there is one that I would like to mention specifically. This publication is called the *Small Computer Systems Journal,* published, edited and put together by a gentleman named Dr. William Schenker. This small journal is not in the professional class, but does have the potential of filling in the holes that some of the bigger books are unable to, particularly in the field of medical applications.

I would suggest that to add to your library of useful information, you write to Dr. Schenker at P.O. Box 6733, Concord, CA 94524. The journal needs your support and will be invaluable to you.

Just recently, we came across a very interesting little club called the *Physicians Microcomputer Club.* This club is, you guessed it, for the medical people who are interested in making the micro work for them. The club president, Dr. Gerald Orosz, informed me that many members have developed the normal business routines for the machines, and are working on using the machine as a diagnostic tool.

For any physician interested in joining a club made up of colleagues we suggest writing to: Dr. Gerald M. Orosz, President, Physicians Microcomputer Club, Box 6483, Lawrenceville, NJ 68648. This is a club that is working to be a national organization and we would like to see them make it.

One of the pleasures that I have as an editor is to visit manufacturers and get a bird's-eye view of what is taking place. This last month I visited the EXTENSYS plant in the San Francisco Bay Area, and talked to Dan Pichulo, their marketing manager.

It seem that EXTENSYS, like so many other manufacturers, has found that the current and most rapidly growing marketplace is small business systems. As a result, EXTENSYS has engineered a system they call the EX 3000, which is designed either as a single user, or multi-user system. The system is not totally earmarked as a business system, but is designed in such a way as to be useful in many applications. One of the more exciting ideas is using the EX 3000 in an educational environment.

The EXTENSYS people plan to exhibit the entire system this year at NCC in Anaheim. Only time will really tell the viability of the system, but from what we saw it looks good.

Also during my travels, I had the chance to visit the folks at Pertec MITS and found out that they are very much interested in the business market. We plan to have an interview with Pertec, covering some of their thoughts and ideas, related to what is happening in the market and where they are headed.

NOTEBOOK

An area that I find it necessary to address this month is book reviews. INTERFACE AGE provides the Book Review column to allow readers to put forth their feeling of a particular book. The Book Review column is like any review section of any publication, its purpose is not to sell the book, but to say 'here it is and in this reader's opinion has this much value.'

Occasionally a book is panned, but this is not to say that the book has no net worth. The book was only panned by one reviewer, who has the right to say what he or she feels as long as it is not malicious in intent. Should an author, or another reader, feel that a review was unfair, we are more than willing to print his or her response in the Letters to the Editor column.

Every time that I go to a seminar, or attend a computer club meeting, one of the first questions that pops up is the standardization of BASIC, followed by information exchange standardization.

Currently there are over 85 versions of BASICs, with each version filling a certain user need. However, for real time useful applications, BASIC will not be the language used, regardless of how powerful a version is created. In my opinion, and I stress that this is my opinion, the worthwhile and cost effective applications packages will be written in assembly code, with everything transparent to the end user.

Now obviously I am not saying that this approach is for every use or user. However, for the small businessman who is only interested in how useful and cost effective the software is that he purchased, assembly language applications will be the best possible choice.

This is not meant to say that BASIC will die, or attempts at standardization will not and are not taking place. They are and will. As a matter of fact, it is felt by many people in the industry that a de facto standard BASIC does exist, that being MITS 12K extended.

Another question that is in line with standardization of BASIC is: Why BASIC at all? After all, there are so many other possible choices that lend themselves to microprocessors. This is most certainly true and only time will tell exactly what will be the new language of the industry. A little later this year we will address this particular aspect of software in depth, and possibly give you a surprise or two.

The second question that I mentioned, which seems to intrigue users, is a standard for information exchange. Such standards have existed for a number of years, but have not been addressed in a manner totally suitable for the micro industry.

INTERFACE AGE is just as guilty as anyone else, and we have evidenced it in our previous Floppy ROMs™. We have tried several methods of implanting the data on the sound sheet, and all have worked, but none have been universal.

Consequently, we embarked upon the unusual idea of already using what was available, and with a little bit of work, making it work with the Floppy ROM and remote terminal operations. The result of this great insight is IAPS™, (International ASCII Publishing Standard). As you read this month's issue, Alan Miller and Bill Turner will introduce the IAPS concept, which we hope will spark even more innovations.

You will probably notice, as you read Alan and Bill's article that the listings are very legible. However, due to the importance of IAPS™ we will make available copies of the desired programs, full size, for anyone sending in a self-addressed stamped envelope and exactly which program or programs you desire. We are only making a limited number of these available, so it will be on a first come first serve basis.

We are very anxious to hear your comments on IAPS and will try very hard to implement all the useful ideas we see. Our goal is to make this one of the more useful software tools around.

As a final note, I would like to request club chairmen, or presidents of computer clubs, to write to us and submit articles about their clubs and activities. We are very anxious to hear from you and would like to publish a guide to clubs sometime in the future. To do this we need to know who is active and where you are.

Please address your letters about your clubs to: Clubs, INTERFACE AGE Magazine, P.O. Box 1234, Cerritos, CA 90701.

—carl

OSBORNE & ASSOCIATES, INC.

The World Leaders In Microprocessor Books

If you want information on microprocessors, begin with the Osborne books.

PROGRAM BOOKS WRITTEN IN BASIC

Payroll With Cost Accounting
Accounts Payable And Accounts Receivable
General Ledger

These books may be used independently, or implemented together as a complete accounting system. Each contains program listings, user's manual and thorough documentation. Written in an extended version of BASIC.
#22002 (400 pages), #23002*, #24002*

Some Common BASIC Programs

76 short practical programs, most of which can be used on any microcomputer with any version of BASIC. Complete with program descriptions, listings, remarks and examples.
#21002 (200 pages)

ASSEMBLY LANGUAGE PROGRAMMING

8080A/8085 Assembly Language Programming
6800 Assembly Language Programming

These books describe how to program a microcomputer using assembly language. They discuss classical programming techniques, and contain simplified programming examples relevant to today's microcomputer applications.
#31003, 32003 (400 pages each)

AN INTRODUCTION TO MICROCOMPUTERS

Volume 0 - The Beginner's Book

If you know nothing about computers, then this is the book for you. It introduces computer logic and terminology in language a beginner can understand. Computer software, hardware and component parts are described, and simple explanations are given for how they work. Text is supplemented with creative illustrations and numerous photographs. Volume 0 prepares the novice for Volume I.
#6001 (300 pages)

Volume I — Basic Concepts

This best selling text describes hardware and programming concepts common to all microprocessors. These concepts are explained clearly and thoroughly, beginning at an elementary level. Worldwide, Volume I has a greater yearly sales volume than any other computer text.
#2001 (350 pages)

Volume II — Some Real Products (revised June 1977)

Every common microprocessor and all support devices are described. Only data sheets are copied from manufacturers. Major chip slice products are also discussed.
#3001A (1250 pages)

PROGRAMMING FOR LOGIC DESIGN

8080 Programming For Logic Design
6800 Programming For Logic Design
Z80 Programming For Logic Design

These books describe the meeting ground of programmers and logic designers; written for both, they provide detailed examples to illustrate effective usage of microprocessors in traditional digital applications.
#4001, #5001, #7001 (300 pages each)

OSBORNE & ASSOCIATES, INC. • P.O. Box 2036 • DEPT. C18 • Berkeley, California 94702 • (415) 548-2805 9:00 a.m. - 5:00 p.m. PST TWX 910-366-7277

Price applies to orders received by June 30, 1978.		PRICE	QTY	AMT
6001	Volume 0 — The Beginner's Book	$ 7.50		
2001	Volume I — Basic Concepts	7.50		
3001A	Volume II — Some Real Products (1977 edition)	15.00		
4001	8080 Programming For Logic Design	7.50		
5001	6800 Programming For Logic Design	7.50		
7001	Z80 Programming For Logic Design	7.50		
31003	8080A/8085 Assembly Language Programming	7.50		
32003	6800 Assembly Language Programming	7.50		
21002	Some Common BASIC Programs	7.50		
22002	Payroll With Cost Accounting	12.50		

- 6-1/2%, SF Bay Area residents only
- 6%, California residents outside SF Bay Area
- Payment must be enclosed for orders of 10 books or less.
- I have enclosed:

☐ check ☐ money order

TOTAL
Sales Tax (Calif. residents only)
Shipping Charges
TOTAL AMOUNT OF PURCHASE

NAME
ADDRESS
CITY STATE ZIP PHONE

SHIPPING CHARGES Shipping charges for bulk orders to be arranged.

☐ 4th class (no charge, allow 3-4 weeks within USA, not applicable to discounted orders)
☐ $.50 per book, UPS (allow 10 days) in the U.S.
☐ $1.50 per book, special rush shipment by air in the U.S.
☐ All foreign orders, $3.00 per book, for air shipment

***These books are scheduled to be published during 1978. Please notify me when they are available:**
☐ 23002 Accounts Payable and Accounts Receivable
☐ 24002 General Ledger

Price increase scheduled for July 1, 1978

Please send information on:
☐ pricing, ordering, and titles available after 6/30/78
☐ initial dealer consignments
☐ dealer and school discounts
☐ foreign distributors

F8

CIRCLE INQUIRY NO. 14

LETTERS TO THE EDITOR

Dear Editor:

In Professor Patt's article, "Programmable Calculator Modeling of Experimental Data" (INTERFACE AGE, February 1978), an algorithm was presented for the fitting of standard probability density funcitons to experimentally obtained statistics. In conjunction with this algorithm, a set of companion-programs was presented for use with the Hewlett-Packard Model HP-67 pocket calculator. The programs perform a best least-squares fit, and display both the curve parameters and the square of the correlation coefficient (goodness-of-fit).

A second set of companion-programs is now presented for use with the Texas Instruments Model TI-59 programmable pocket calculator. Like the Hewlett-Packard programs, a block-structure format has been selected for compactness of presentation.

GENERAL PROGRAM TO MODEL EXPERIMENTALLY OBSERVED RANDOM BEHAVIOR

USING A TI-59 PROGRAMMABLE POCKET CALCULATOR

(Program begins in location 000)

```
STO 07  Pgm 01  SBR CLR  R/S
STO 08
Lbl A'  2 + RCL 03 =  R/S
STO 09  CLR  R/S
STO 10
INSERT # 1
Σ+  GTO A'
Lbl A  OP 12
INSERT # 2
STO 07  GTO C'
```

```
Lbl B  OP 12  x⇄t
INSERT # 3
STO 08
Lbl C'  R/S
INSERT # 4
STO 09
Lbl D'  R/S
INSERT # 5
GTO D'
```

Normal Density Function

```
Insert #1:  ÷ RCL 08  = ln ÷ (RCL 07 - RCL 09  = x⇄t RCL 09 + RCL 07
            = ÷ 2 =
Insert #2:  (not used)
Insert #3:  √x
Insert #4:  RCL 08  * 2 √x * π √x = STO 10  OP 13  x^2
Insert #5:  - RCL 07  = ÷ RCL 08  = x^2 +/- INV lnx √x ÷ RCL 10 =
```

Log-Normal Density Function

```
Insert #1:  ÷ RCL 08  * RCL 09  ÷ RCL 07  = lnx ÷ (RCL 07  ÷ RCL 09)
            lnx =  x⇄t RCL 07  * RCL 09  = √x ln
Insert #2:  (not used)
Insert #3:  √x
Insert #4:  RCL 06  * 2 √x  * π √x  = STO 10  OP 13  x^2
Insert #5:  - x⇄t RCL 07 = ÷ RCL 08  = x^2 +/-  INV  lnx √x  ÷
            x⇄t ÷ RCL 10 =
```

Exponential Density Function

```
Insert #1:  x⇄t  RCL 09  - RCL 07  = x⇄t  1/x  * RCL 08  = ln
Insert #2:  RCL 06  ÷ RCL 05  =  1/x
Insert #3:  0  1/x
Insert #4:  RCL 06  x^2  ÷ RCL 05  ÷ RCL 02  =
Insert #5:  +/-  ÷  RCL 07  = INV  lnx  ÷ RCL 07  =
```

Cauchy Density Function

```
Insert #1:  1/x  -  RCL 08  1/x  =  1/x  * π *  (RCL 09 - RCL 07)*
            x⇄t  2 =  x⇄t  * (RCL 09 + RCL 07 =
Insert #2:  (not used)
Insert #3:  (not used)
Insert #4:  OP 13  x^2
Insert #5:  x^2 +  RCL 07  x^2 =  1/x  *  RCL 07  ÷ π  =
```

Rayleigh Density Function

```
Insert #1:  x⇄t  RCL 09  x^2  -  RCL 07  x^2 =  x⇄t  1/x  * RCL 08 *
            RCL 09  ÷ RCL 07  =  x^2  ln
Insert #2:  RCL 06  ÷  RCL 05  =
Insert #3:  0  1/x
Insert #4:  RCL 06  x^2  ÷ RCL 05  ÷ RCL 02  =
Insert #5:  * (x^2 +/-  * RCL 07) INV lnx √x  * RCL 07  =
```

Gamma Density Function

```
Insert #1:  RCL 07  ÷ RCL 09  = ln 1/x  *  x⇄t  (RCL 09 - RCL 07 ) =
            x⇄t  * (RCL 08 ÷ RCL 10 ) ln =
Insert #2:  (not used)
Insert #3:  1/x
Insert #4:  RCL 07  +  1 = E  * RCL 08  y^x  (RCL 07 + 1 = STO 10
            OP 13  x^2
Insert #5:  ÷ x⇄t  RCL 08  +/-  = INV lnx ÷ RCL 10  * x⇄t  y^x
            RCL 07  =
Subrtne E:  Lbl E  ( ( CE + 3 ) ( STO 00  * ( 1/x  x^2 * (CE * (CE *
            ( +/-  * 1.5 + 2 )
            ÷ 7 - 1 ) ÷ 30 + 1 ) ÷ 12 - 1 ) + ( RCL 00 + .5 ) *
            RCL 00  lnx )
            INV lnx * 2 √x  * π  √x ÷ RCL 00  ÷ ( 2 + RCL 00 *
            ( CE - 3 ) )
            ÷ ( RCL 00  - 3 ) ) INV SBR
```

Beta Density Function

```
Insert #1:  RCL 07  ÷ RCL 09  =  lnx  1/x  *  x⇄t  ( ( 1 - RCL 07 ) ÷
            ( 1 - RCL 09 ) ) ln  =  x⇄t  *  ( RCL 08  ÷ RCL 10 ) ln  =
Insert #2:  + 1 =
Insert #3:  + 1 =
Insert #4:  RCL 07  + RCL 08  =  E  ÷
            RCL 07   E  ÷
            RCL 08   E  =  STO 10  OP 13  x^2
Insert #5:  y^x  x⇄t  ( RCL 07  - 1 ) * ( 1 - x⇄t ) y^x (RCL 08  - 1 )
            * RCL 10 =
Subrtne E:  (same as subroutine E used with Gamma density function)
```

L. Fisher, J.R.A. Lemieux, & M. Patt
University of Lowell

Dear Editor:

This is a letter in response to your Survey for Floppy ROM #3 (January, 1978).

Since most of the questions do not apply to me and I like my magazine with all the pages whole, I wrote this letter.

I bought the magazine on a book rack at the Olson Electronics store. It looks fine.

I could not use this system because I don't have the specific equipment used. But I *do* like the concept.

As for future programs: how about programs to "translate" other programs across industry "standards" (i.e. MITS BASIC to IMSAI BASIC, etc. [even MITS 4.0 to 4.1, etc.]); and programs to cover a variety of handling mediums to increase versatility (like driver routines for several KC standard formats and many other similar things); also, paper tape reader formats, modem formats, etc.; home "handling" programs, for heating, lighting, phone answer, etc.; cassette file system for the poor home/business system (AUDIO).

It's highly recommended that more thought go into making these more general. This word processor only works on a specific hardware design and with a specific existing software package! Not to mention a fairly costly system at that.

Also, in the same vein, try to wait a little longer, so the program is really done. That is, one that is easily modified with parts clearly identified for deletion, patching, additions, and relocations. I know it's tough, but it's got to be that way to be general and not just "one man's answer to a problem."

Oh, label the Floppy ROM™ with format and such too, as it becomes separated from text for use or redistribution.

Gary L. Camp
Pomona, CA

Gary, you must be reading our minds. We are addressing the areas that you have mentioned. We are making every effort possible to ensure the viability of the Floppy ROM™, and you should find that the

one contained in this issue is very, very usable.
Yes, we are working on file systems, handling programs, and somethings we just don't want to mention quite yet.

Dear Editor:

I am still enjoying your magazine. It is very interesting because I like computers. Please keep on writing all those good articles.

Christopher L. Bailey
Orlando, FL

Thanks.

Dear Editor:

Your articles are excellent, but I was not able to find any reference where one might be able to obtain the "Medical Accounts Receivable Package," which was described in your January, 1978 issue.

Does your magazine reproduce individual articles? If so, kindly advise me how I can obtain them.

Vernard L. Price
Black Creek, WI

Mr. Price, yes we reproduce articles, but require a minimum order. To obtain information on the package write to Mal Lockwood, Administrative Systems Inc., 222 Milwaukee, Suite 102, Denver, CO 80206.

Dear Editor:

I would like to find out how home computers could be used to help the work of district agents for a particular life insurance company, which I plan to work for. I am currently enrolled in a computer management course.

Could you please suggest articles or other research materials which might be helpful to me? Could you please include information as to institutional subscribers to your publication in my area which might aid me in my research?

I plan to submit my findings to the national sales director for the company involved. Thanks for your help.

Robert H. Gill
Berkeley, CA

Readers, please address your answers to INTERFACE AGE. We will forward them to Mr. Gill.

Dear Editor:

The execution time of a program is of concern to many microcomputer applications. One program in your magazine written by Ed Hughot to do a half-hour Monte Carlo simulation (January 1978, "The Use of Microcomputers in Business Risk Analysis") contains his solution to the questions of how long did the program run, did the "fix" speed or slow execution, and how much should I charge for this run.

Readers interested in evaluating his program will be interested to note that it assumes that a Lincoln Semiconductor uCT-1 Time Card resides at port 60H. For a complete description of the Time Card and how to use it, (as well as a copy of Ed's BASIC subroutines), please write:

Lincoln Semiconductor
P.O. Box 68
Milpitas, CA 95035
(408) 734-8020.

Larry A. Lincoln
President

Dear Editor:

I read INTERFACE AGE each month and think it is a fine magazine. However, I have some comments.

In the February, 1978 issue you have an article on modeling of experimentor data with a programmable calculator. In the article it is mentioned that programs for both H-P and TI calculators are given. But only the program for the HP-U7 was printed.

In the letters column of the same issue you mention that you will be publishing a Floppy ROM™ of games in August. More than likely this will not be compatible with my KIM-1 (6502) based system.

Have you considered developing a "Floppy ROM BASIC" which you could make available for the major chips (8080, Z80, 6502, 6800). Then all Floppy ROM applications could be compatible.

H. Thompson
Kenmore, NY

Just keep watching.

THE 16-BIT MICRO

Introducing the AM-100™, an S-100 bus compatible 16-bit microprocessor board set replacing 8-bit based processors. This CPU two board set offers multi-user/multi-tasking timesharing in an advanced disk operating system. It provides 16-bit flexibility and speed with 11 digit floating point arithmetic while supporting many of the economical S-100 bus peripherals such as static memory, memory paging and I/O facilities.

COMPLETE DEVICE INDEPENDENCE

Utilizing Western Digitals WD-16 processor, the AM-100™ incorporates a real-time clock, offers complete device independence, disk file management system and utilities, multi-user structured file system with pass words, extended compiler and reentrant software, multi-level DMA and vectored interrupt system. For large timesharing applications, the AM-100™ not only can utilize hard disk storage and access (AM-400™, AM-500™), but also has incorporated a memory management system allowing partitions up to 64k-bytes.

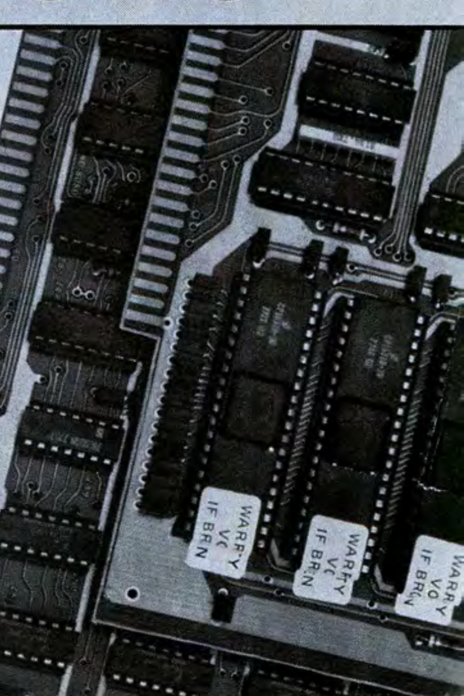
SOFTWARE INCLUDED

All software is licensed to the board set as part of the system. The AM-100™ two board CPU set includes along with its operating system AMOS™ a multi-pass Macro-Assembler, ALPHABASIC™ compiler, ALPHALISP™, ALPHAFORTH™, ALPHA-PASCAL™, SORT, ISAM and other various utilities.

Write or call us for the location of your nearest Alpha Microsystems Dealer.

ALPHA MICROSYSTEMS™

17875N Sky Park North
Irvine, California 92714
Phone: (714) 957-1404



CIRCLE INQUIRY NO. 1

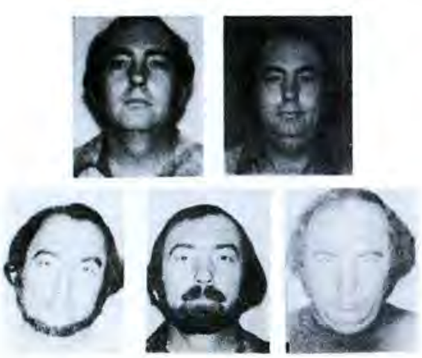
UPDATE

POLICE BULLETIN

In early 1977 the DataSync Corporation was started by Col. David Winthrop and advertised extensively in the national computer magazines. In late June 1977 police detectives learned that the Colonel had defrauded a Santa Maria man of $10,000 under the pretense of designing a computer board for him. The police learned that Col. Winthrop had used other names and had obtained drivers licenses under at least three names. Detectives obtained an Arrest Warrant for Winthrop along with Search Warrants for his home and his business. Winthrop was arrested at his business and was held under $100,000 bail.

A check of Winthrop's fingerprints revealed that his true name was Norman Henry Hunt Jr. and that he had been a parole violator in California since 1965. Hunt was charged with three counts of false pretense theft as felonies and he entered a guilty plea to the charges. Hunt was sentenced to two years, eight months in prison on the Santa Maria charges. Although the investigation revealed that Hunt had been involved in fraud and false pretense thefts amounting to a quarter of a million dollars in five different states within the past four years, other local, state and federal agencies declined to prosecute Hunt.

On February 26, 1978, Hunt, who had been serving his term at Chino State Prison, escaped from a minimum security facility.



Hunt is currently being sought by California authorities. Hunt is a white male, 6'3", 220 lbs., reddish-brown hair, hazel eyes. He has worked as a TV repairman, and has run businesses marketing CB radios, computer products and jacks for trailers. His method of operation has been to move to a town under a new identity, rent a house with option to buy, and to make contacts in his field of endeavor (recently, *computer hobbyists*).

Hunt will generally begin his operation by soliciting backing for product design from private parties. Often he will sell his qualifications so well that it is the victim's idea to ask Hunt to design a product for Hunt.

Hunt may then start a business and solicit partners. He will rent a building, hire employees, begin a credit line with suppliers. After enough equipment has been received from suppliers on credit to look impressive, he will apply for a bank loan to start production. He will usually go to a local bank rather than a large bank chain. If the loan is received, Hunt empties the business of its equipment and leaves the area, leaving the creditors and the bank high and dry.

Hunt also orders equipment from dealers and pays by check. He then stops payment on the checks. When contacted by a dealer about why he stopped payment, he may say that

the product was defective and is being returned, and request that another unit be shipped.

When Hunt was arrested in Santa Maria, he had a loaded shotgun in his closet at home, and he may be considered dangerous. If anyone has information that may relate to this suspect, please contact Detective Ernest L. Kapphahn, Santa Maria Police Department, (805) 922-7811, or Investigator Zeke Hernandez, Chino State Prison, (714) 597-1821.

MICRO EXPO 1978

Micro Expo 1978, sponsored by Sybex, will be held in Paris, France, May 23-25.

For more information, in Europe write to 313 rue Lecourhe, 75015 Paris, telephone: (1) 828-2502; in the U.S.A. write 2161 Shattuck Ave., Berkeley, CA 94704, (415) 848-8233.

PERSONAL COMPUTING '78

Personal Computing '78 will be held at the Philadelphia Civic Center August 24-27, 1978.

August 24 is the first full-day Industry Trade Show, which is open to dealers, the industry, and exhibitors' guests. Special meetings and seminars for dealers and retailers are planned for this evening at convention headquarters.

August 25-26-27 the full Personal Computing Show and Personal Computing College will be running. Over 80 hours of free seminars are planned. Professional seminars featuring in-depth study will be conducted during the week by companies such as Adam Osborne & Associates, Sybex, and Tychon, Inc. at the nearby Hilton Hotel.

PC '78 will be the largest and longest Personal Computing Show yet held, with over 300 booths and running four days.

Philadelphia is centrally located within the largest computer market in the United States, with 21.4% of the market within an easy two hour drive.

For more information contact John H. Dilks III, Rtl. Box 242 Warf Rd., Mays Landing, NJ 08330, (609) 653-1188.

MIDWEST PERSONAL COMPUTING SHOW RETURNS TO CHICAGO

The Midwest Personal Computing Exposition makes its second annual appearance in Chicago, October 5-8, 1978 at the Apparel Center's Expocenter (directly across the street from the Merchandise Mart).

It will be the only 1978 personal computing exposition in Chicago officially sponsored by Personal Computing magazine. This year the event will be produced by Industrial and Scientific Conference Management, Inc. of Chicago.

Last year nearly 13,000 visitors jammed the exhibit floor for the inaugural show to see and buy the equipment, accessories and software. All sectors of society were represented — businessmen, educators, doctors, lawyers, computer professionals, students, hobbyists, and home users.

More than 200 displays featuring the full spectrum of latest personal computing developments are expected to be presented by America's top manufacturers and distributors. A comprehensive program of seminars, forums and practical-application clinics will parallel the four days of exhibits.

For complete exhibitor and visitor details contact Midwest Personal Computing Exposition, ISCM, 222 W. Adams St., Chicago, IL 60606, (312) 263-4866.

RADIO SHACK SETS SIGHTS ON #1 POSITION IN MICROCOMPUTING

Radio Shack reaffirmed its intentions to be serious, progressive, aggressive, and number one in the market, with their TRS-80 Microcomputer System. In entering the microcomputer field, Radio Shack's intention is to explore and enter the market for computers to serve small business, small parts of large business, labs, schools, professional people, students and, yes, hobbyists.

Radio Shack has the know-how, the financial muscle to do this; and perhaps, most importantly, almost exactly 7,000 locations to display and sell TRS-80 equipment, peripherals, software and systems.

Radio Shack has introduced a new Basic Level-II expansion interface and ROM converter, mini-floppy disk drive unit, line printer, new software and complete TRS-80 expanded systems selling for up to $4,000.

Radio Shack introduced their TRS-80 Microcomputer System in August 1977. The Level-I Basic 4K system, which sells for $599.00 completely wired and tested, ready to plug in and use, consists of a 53-key professional-type keyboard and microcomputer plus regulated power supply, a computer-controlled data cassette recorder, and a 12" video display monitor.

A comprehensive owner's manual is supplied with the TRS-80 that explains everything necessary for its operation from plugging it in through programming.

Radio Shack supplies pre-recorded cassette programs for such applications as a small business payroll and personal finance management. Educational programs include math, algebra, and even a Level-I basic computer programming course.

Radio Shack, a division of Tandy Corporation (NYSE), is headquartered in Forth Worth, Texas which is also where they manufacture the TRS-80.

NEW MAN AT PERTEC

Donald L. Tollefson has been appointed Western Regional Field Sales Engineer for the Pertec Division of Pertec Computer Corporation.

In his new capacity, Tollefson will be responsible for developing new OEM customers as well as coordinating existing customers. He will report to Bert Johnston, Western Regional Sales Manager. He replaces Jeff Segers, who has transferred to Pertec's Microsystems Division.

Tollefson comes to Pertec with 15 years of experience in the peripheral and instrumentation sales field. Prior to joining Pertec, he was Sales Manager for Data Processing Design, Inc., Buena Park, CA. Previously he served as Western Regional Sales Manager for Diva, Inc., Eatontown, NJ; and prior to that had held positions with Electronics Marketing Specialists, Inc.

He is a member of the Precision Measurement Association and the Instrument Society of America. He received a BSEE degree from UCLA.

COMPUTERFEST™ '78

The 3rd Annual MACC Computerfest '78 will be held June 23-25 in Detroit, sponsored by the Midwest Affiliation of Computer Clubs.

The MACC is holding the 1978 Computerfest in the Renaissance Center — a $500 million total environment complex one-third larger than Rockefeller Center.

Featured will be hobbyist exhibits, tours and evening activities, club hospitality suites, special club meetings, fabulous programs, manufacturers party, technical sessions, giant flea market, free seminars, new products and exhibits.

For more information write MACC Computerfest, Box 9578, Dept. LIT, Detroit, MI 48202, or call the 24-hour Hotline number (313) 775-5320.

CALL FOR PAPERS

The 11th Annual Microprogramming Workshop will be held November 19-22, 1978, at the Asilomar Conference Ground in Pacific Grove, California.

Papers on the following topics are solicited:

- Microprogramming for security, reliability and testing
- Microprogramming in multiprocessor systems
- Hardware for emulation
- Firmware engineering systems
- Microprogrammed LISP machines
- Machine descriptions
- Microprogrammable special-purpose architectures
- Microprogramming I/O functions
- Signal processing applications
- Microprogramming bit slice architecture

Outstanding papers not directly related to the above topics are also invited.

This workshop will provide a forum for the discussion and comparison of design techniques for firmware and for the supporting hardware. Informal interaction between groups working in similar research and application environments will highlight the Topical Sessions.

Panel discussions will be held on selected topics, including: User microprogrammable machines; LSI circuits and the future of microprogramming; third generation microprogrammed processors-What we have learned; designers notebook of micro-

Why Apple II is the world's best selling personal computer.

Which personal computer will be most enjoyable and rewarding for you? Since we delivered our first Apple® II in April, 1977, more people have chosen our computer than all other personal computers combined. Here are the reasons Apple has become such an overwhelming favorite.

Apple is a fully tested and assembled mainframe computer. You won't need to spend weeks and months in assembly. Just take an Apple home, plug it in, hook up your color TV* and any cassette tape deck — and the fun begins.

To ensure that the fun never stops, and to keep Apple working hard, we've spent the last year expanding the Apple system. There are new peripherals, new software, and a 16-chapter Owner's Manual on "How to Program in BASIC." There's even a free Apple magazine to keep owners on top of what's new.

Apple is so powerful and easy to use that you'll find dozens of applications. There are Apples in major universities, helping teach computer skills. There are Apples in the office, where they're being programmed to control inventories, chart stocks and balance the books. And there are Apples at home, where they can help manage the family budget, control your home's environment, teach arithmetic and foreign languages and, of course, enable you to create hundreds of sound and action video games.

When you buy an Apple II you're investing in the leading edge of technology. Apple was the first computer to come with BASIC in ROM, for example. And the first computer with up to 48K bytes RAM on one board, using advanced, high density 16K devices. We're working to keep Apple the most up-to-date personal computer money can buy. Apple II delivers the features you need to enjoy the real satisfaction a personal computer can bring, today and in the future.

15 colors & hi-resolution graphics, too.

Don't settle for a black and white display! Connect your Apple to a color TV and BASIC gives you instant command of three display modes: Text, 40h x 48v Color-graphics in 15 colors, and a 280h x 192v High Resolution array that lets you plot graphs and compose 3-D images. Apple gives you the added capability of combining text and graphics, too.

Back to basics, and assembly language too.

Apple speaks three languages: fast integer BASIC, floating point BASIC for scientific and financial applications, and 6502 assembly language. That's maximum programming flexibility. And, to preserve user's space, both integer BASIC and monitor are permanently stored in 8K bytes of ROM, so you have an easy to use, universal language instantly available. BASIC gives you graphic commands: COLOR=, VLIN, HLIN, PLOT and SCRN. And direct memory access, with PEEK, POKE and CALL commands.

Software: Ours and yours.

There's a growing selection of preprogrammed software from the Apple Software Bank — Basic Finance, Checkbook, High Resolution Graphics and more. Now there's a User Section in our bank, to make it easy for you to obtain programs developed

CIRCLE INQUIRY NO. 2

by other Apple owners. Our Software Bank is your link to Apple owners all over the world.

Alive with the sound of music.

Apple's exclusive built-in speaker delivers the added dimension of sound to your programs. Sound to compose electronic music. Sound to liven up games and educational programs. Sound, so that any program can "talk" back to you. That's an example of Apple's "people compatible" design. Another is its light, durable injection molded case, so you can take Apple with you. And the professional quality, typewriter-style keyboard has n-key rollover, for fast, error-free operator interaction.

Apple is the proven computer.

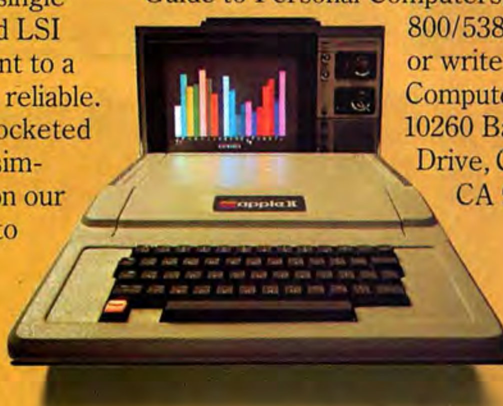
Apple is a state-of-the-art single board computer, with advanced LSI design to keep component count to a minimum. That makes it more reliable. If glitches do occur, the fully socketed board and built-in diagnostics simplify troubleshooting. In fact, on our assembly line, we use Apples to test new Apples.

Apple peripherals are smart peripherals.

Watch the far right column of this ad each month for the latest in our growing family of peripherals. We call them "intelligent interfaces." They're smart peripherals, so you can plug them in and run them from BASIC without having to develop custom software. No other personal computer comes close to Apple's expandability. In addition to the built-in video interface, cassette I/O, two A/D game paddles, and two more A/D inputs, Apple has eight peripheral slots, three TTL inputs and four TTL outputs. Plus a powerful, state-of-the-art switching power supply that can drive all your Apple peripherals, including two disks.

Available now.

Apple is in stock and ready for delivery at a store near you. Call us for the dealer nearest you. Or, for more details and a copy of our "Consumer Guide to Personal Computers," call 800/538-9696 or write Apple Computer, Inc., 10260 Bandley Drive, Cupertino, CA 95014.



New from Apple.

Introducing the Apple Communication Interface

Apples of the world unite! Now you can, with our new intelligent communication interface card. Just plug it in and it turns your Apple into an intelligent terminal that can go on line with other terminals, time-sharing computers and, especially, with other Apples. You can even play Tele-Pong! Everything you need is on one small card. With a modem, it enables your Apple to communicate by phone at 110/300 baud RS232 full duplex I/O. The card is fully assembled and tested and has all required software in on board ROM. It's controlled by simple BASIC commands. And it's available from stock.

Peripherals in stock

Hobby Board, Parallel Printer Interface, Communication Interface.

Coming soon

High speed serial printer interface, General purpose serial interface, Printer II, Printer IIA, Disk II, Monitor II.

* Apple II plugs into any standard TV using an inexpensive modulator (not included).

apple computer®

Programming is a snap! I'm halfway through Apple's BASIC manual and already I've programmed my own Star Wars game.

Those math programs I wrote last week–I just rewrote them using Apple's mini-assembler and got them to run a hundred times faster.

Apple's smart peripherals make expansion easy. Just plug 'em in and they're ready to run. I've already added two disks, a printer and the communications card.

programming techniques and design of microprogrammable hardware.

Suggestions for panel session topics, participants, and formal paper session subjects are welcomed. Tutorial, survey, and evaluation papers are invited, along with papers describing specific research results.

Papers should be submitted in triplicate to:

Dr. Alice C. Parker
MICRO-1 Program Chairman
Dept. Electrical Engineering
Carnegie-Mellon University
Pittsburgh, PA 15213
(412) 578-2472

Deadline for formal paper submission is June 1, 1978.

QUANTOR ANNOUNCES XEROX RESEARCH AGREEMENT

Quantor Corporation has signed a research and development contract with the Data Systems Division of Xerox Corporation in El Segundo, California.

The agreement calls for Quantor to test the feasibility of integrating its microfilm technology with a Xerox product development program.

First stage of feasibility testing will be funded by an initial payment from Xerox of approximately $50,000. If fully completed, the program could run 18 months and involve Xerox expenditures up to $450,000. Xerox has reserved the right to withdraw from the agreement at several milestone points.

The agreement also has an option under which Xerox may engage Quantor to manufacture a product developed under the research and development contract.

Quantor, headquartered in Mountain View, California, is a leading manufacturer and supplier of COM systems and related equipment.

INTERNATIONAL BANKS ORDER ON-LINE COMPUTER SYSTEMS

A number of overseas financial institutions have begun implementing on-line computer systems using NCR equipment.

Included is the Societe Generale Bank, the first bank in Nigeria to install an on-line system. The Societe Generale network is based on two NCR I-8250 computers. Other equipment includes six NCR 299 accounting computers and four NCR 7200 data-entry subsystems.

In Taiwan, the Chang Hwa Commercial Bank, is installing an on-line system linking five cities. The system uses an NCR V-8570 computer and NCR teller terminals. In addition to on-line deposit and withdrawal services, the bank plans to provide automatic account transfer for tax and utility payments. Eventually the bank will establish a central information file system in which all customer information is grouped together in a single master file.

In Berne, Switzerland, the Gewerbekasse Bank has also ordered a V-8550 computer and 31 NCR terminals. Previously, the bank's processing was done by an independent data center.

HARVARD UNIVERSITY ANNOUNCES INTERNATIONAL COMPUTER GRAPHICS WEEK

Harvard University has announced plans for an international Computer Graphics Week July 23-28, 1978, to be sponsored by the school's Laboratory for Computer Graphics and Spatial Analysis.

The event will focus on the Laboratory's International User's Conference on Computer Mapping Software and Data Bases: Application and Dissemination. At the conference over 100 speakers and numerous exhibits from the commercial, educational and governmental sectors will show how computer mapping is being used in city and regional planning, social services, public safety, transportation and engineer-

ing, ecology and the environment, energy, public health, marketing, research and development, management information systems and university research and instruction.

In addition there will be an in-depth review of currently available computer mapping software and data bases, as well as sessions on thematic map design principles and a hands-on workshop at the Harvard Laboratory.

Among special features will be a sesion on software and data base distribution and marketing and an executive briefing seminar to discuss the relevance and projected impact of computer mapping in the commercial sector. For more information, contact Ira Alterman at the Center for Management Research, Executive Plaza, 850 Boylston St., Chestnut Hill, MA 02167, (617) 738-5027.

JOHN VINKE JOINS COMPUTER SCIENCES AS ASSISTANT CONTROLLER

John Vinke has joined Computer Sciences Corporation as assistant controller. He will be responsible for all of the company's corporate accounting and financial reporting functions.

Prior to joining Computer Sciences, Vinke was director of internal audit for Lear Siegler, Inc. of Santa Monica, California.

He was previously associated with public accounting firms for nine years, including seven years with Arthur Andersen & Co., where he served as audit manager. He is a member of the California Society of Certified Public Accountants.

Headquartered in El Segundo, California, Computer Sciences designs and develops complete computer-communications systems and operates an international computer network service called Infonet.

INTEGRATED COMPUTER SYSTEMS

An organization whose sole business is continuing education in high technology fields, has scheduled the following intensive courses:

Fiber Optic Communication Systems
#440 - Four Days

Los Angeles	May 16-19
Toronto	May 30-June 2
Boston	July 11-14
Washington, D.C.	July 25-28

Modern Methods of Digital Signal Processing
#412 - Four Days

Los Angeles	May 2-5
Washington, D.C.	May 9-12
Toronto	June 6-9
Boston	June 20-23

Synthetic Aperture Radar (SAR) Systems
#475 - Three Days

Washington, D.C.	May 19-12
Toronto	July 19-21
Los Angeles	July 26-28

Tuition fees are $595 for each 4-four day course (#440 or #412) and $495 for the 3-day course (#475). Fees include lectures, extensive course materials, luncheons and coffee breaks. Team/group discounts are available for 3 or more attendees from the same company.

For details and free brochures, contact Integrated Computer Systems, Inc., 3304 Pico Blvd., P.O. Box 5339, Santa Monica, CA 90405. (213) 450-2060.

COMPUTERS AND THE AIR FORCE

Computer Sciences Corporation of El Segundo, California, has received a $4.4 million follow-on contract from the U.S. Air Force for studies of the nuclear survivability and vulnerability of military command and control communications, and computer systems.

The 34-month contract will be managed by the company's Albuquerque operations office for the Air Force Weapons Laboratory at Kirtland Air Force Base, NM.

Under the contract, CSC is responsible for planning, performance and integration of the survival studies. CSC heads a team of companies that is analyzing, testing and developing ways to deal with the effects

 CIRCLE INQUIRY NO. 39

MICROMATION

MACRO DISK EXP

THE GROWTH SYSTEM

Micromation introduces the EXP series of expandable floppy disks. Systems designed to be reliable, versatile, and to expand with the needs of your system.

524 Union Street
San Francisco, CA 94133
415-398-0289

STEP UP TO DOUBLE-HEADED

New double-headed drives record data on both sides of a diskette — But no manufacturer is delivering them yet. When these new double-headed drives become available, single headed drives will be obsolete—except for the EXP series. Because Micromation guarantees to exchange any EXP series single-headed drive for a double-headed drive—at a cost of only $300 per drive.

You can also upgrade your system to double density in the third quarter with our double density controller conversion for only $300.

So the dual drive system that you buy now has a capacity of over 500 K Bytes, but can be increased to a capacity of two megabytes. Your system will grow with new technology—not be obsoleted by it.

SOLID SOFTWARE SUPPORT

The Micromation disk controller features IBM 3740 compatibility—and the proven CP/M* operating system. You can also choose between high level languages such as BASIC AND FORTRAN or complete business application and word processing packages.

A COMPLETE, ASSEMBLED SYSTEM

All Micromation systems are fully assembled and tested. There's even a serial I/O port on the controller to make it easy to bring the system up. Just connect your terminal to the serial port, install in any 16K S-100 system, jump to the on-board PROM bootstrap—and you're up and running without any patching. Or you can choose optional I/O drivers for the SOL computer.

EXP systems are complete—including drives, S-100 controller, power supply, and handsome Scandinavian style wood and metal enclosure.

RELIABILITY

EXP drives are manufactured by Memorex, the oldest independent manufacturer of floppy drives. And we stand behind them. All Micromation EXP series products are warranteed for a full six months. An optional one year warranty extension is also available.

UNDER $2,000 SYSTEMS

A complete dual-drive system is available for under two thousand dollars. And there are no high priced options—write protect and front panel activity light are standard. And since the EXP series is based upon proven components and the drives are manufactured by a major supplier, the systems are available in less than four weeks from receipt of your order.

EXP-1 Single drive system	**$1,195.**
EXP-2 Dual drive system	**1,895.**
CP/M with BASIC	**95.**

COMPLETE COMPUTER SYSTEMS

Micromation's EXPSYS is a complete computer system. It includes a Z-80 processor with 32K of memory, serial I/O ports and EXP series drives. An optional video interface, monitor, and keyboard are also available. An S-100 mother board allows the system to be configured to your custom requirements. Yet prices for this complete dual-drive computer system start at less than $4,000. The systems are fully assembled and tested with our full warranty and upgrade policy, of course.

EXPSYS-1 Single drive 32K Z-80 computer	**$3,295.**
EXPSYS-2 Dual drive 32K Z-80 computer	**3,995.**

Micromation dealers are listed on the adjacent page. Check the list for dealer nearest you or ask at your favorite computer store.

*CP/M is a trademark of Digital Research

on communications equipment of surges of electromagnetic energy released by nuclear explosions. These surges, or electromagnetic pulses, can disrupt and damage unshielded communications equipment in ways resembling the effect of a lightning bolt.

Other nuclear effects such as radio propagation disturbances, direct radiation, and ground and air motion will also be considered.

CSC has been performing these studies under various Air Force contracts since 1972.

Computer Sciences engineers and develops computer-communications systems, manages clients' computer facilities, and operates an international network time-sharing service called Infonet.

ACM SEEKS NOMINATIONS FOR ANNUAL GRACE MURRAY HOPPER AWARD

The Association for Computing Machinery is seeking nominations for its Grace Murray Hopper Award, given each year to the outstanding young computer professional selected on the basis of a single recent major technical or service contribution to the computer industry. In order to qualify, candidates must have been 30 years of age or less at the time the qualifying contribution was made.

The Award will be presented at the opening session of the Association's Annual Conference on December 4, 1978, in Washington, D.C. The Award is in the amount of $1,000, donated by the Univac Division of Sperry Rand, and is accompanied by a certificate.

While the Award is given to the outstanding young "computer" professional, emphasis for the 1978 award will be placed on contributions in the fields of business data processing and personal computing. The Committee felt that these fields have not been adequately rewarded for outstanding contributions in the past.

The last three winners of the Grace Murray Hopper Award are: Edward A. Shortliffe, for his development of a program that consults with physicians about diagnosis and treatment of infection; Allen L. Scherr, for his pioneering study in quantitative computer performance analysis; and George N. Baird for the development and implementation of the U.S. Navy's COBOL compiler evaluation system.

Nominations, which may be made by the nominees themselves, should be sent to:

Richard G. Canning
Chairman, ACM Grace Murray
Hopper Award Committee
925 Anza Avenua
Vista, California 92083

In order to be considered for the 1978 Award, nominations should be received no later than June 30, 1978. Please include the following information:

1. Name, address, and phone number of the person making the nomination.
2. Name, address, and phone number of the nominee.
3. A statement (200 to 500 words) on why the candidate deserves the Award, describing the contribution.
4. The date of birth of the nominee and the date on which the qualifying work was completed.

PCC AND SECOINSA SIGN MAJOR AGREEMENT IN SPAIN

Sociedad Espanola de Communicaciones e Informatica S.A. (SECOINSA) and Pertec Computer Corporation (PCC) have signed an agreement for manufacturing/distribution of PCC data entry systems products in Spain, Portugal and certain North African countries.

Under the agreement, SECOINSA — a publicly held company created by leading members of Spain's data processing, communications, banking and government communities — will distribute computer system data entry products being manufactured at PCC's facilities in Santa Ana, California. Those systems are in the XL family, the most powerful of which is the XL40 Distributed Key Processing® System.

This agreement, the first signed with a U.S. computer firm, represents a major opportunity for the exchange of technical and marketing opportunities between the two companies. SECOINSA will concentrate at the outset on marketing XL systems, then eventually could provide engineering and manufacturing support at SECOINSA facilities in Spain.

SECOINSA maintains headquarters in Madrid and employs more than 630 persons throughout Spain. The total market for computer system sales and services in Spain last year exceeded $375 million.

Pertec Computer Corporation designs, manufactures, markets and services digital magnetic tape transports, disk drives, flexible disk drives and small computer systems under the brand names of iCOM, MITS/Altair, XL40 and CMC.

CALENDAR

May 30 Southern California Users of RT-11 (SCURT) will meet at 9:30 A.M. at USC's Annenberg School of Communications. For details call Mark Bartelt, (213) 795-6811, ext. 2663; or Ray Rittenhouse, (213) 640-1830, ext. 225.

June 1 Microcomputer Users Group (MCG) will hold its meeting at the University of Minnesota, Electrical Eng. Rm. 115 at 7 P.M. The club meets every Thursday. For more information write MCG, Dept. of Elec. Eng., 123 Church St. S.E., Minneapolis, MN 55455.

June 1 Bay Area Microprocessors Users Group (BAMUG) will meet in the Hayward ROC Center, 26316 Hesperian Blvd., Hayward, CA at 7:30 P.M. For further details write BAMUG, 1211 Santa Clara Avenue, Alameda, CA 94501.

June 2 Crescent City Computer Club will hold its meeting at the University of New Orleans, Lakefront Campus at 8 P.M. Call Bob Latham at (504) 722-6321 for more details.

June 3 Louisville Area Computer Club (LACE) will meet at the University of Louisville, Speed School Auditorium at 1 P.M. For details, write the club at 115 Edgemont Dr., New Alban, IN 47150.

June 3 The Computer Hobbyist Group will meet at 1 P.M. in the Green Center, Rm 2.530, of Univ. of Texas, Dallas. For details write to P.O. Box 11344, Grand Prairie, TX 75051.

June 3 South Central Kansas Amateur Computer Association, 9:00 A.M., Wichita Public Library, Wichita, KS. For further information call Chris Borger at (316) 265-1120 or Dave Rawson, 1825 Gary, Wichita, KS 67219, (316) 744-1629 for further details.

June 3 Oklahoma Computer Club will be meeting at the Belle Aisle Library at 10 A.M. Call Al Campbell at (405) 842-4933 for details.

June 3 Milwaukee Area Computer Club will meet at 1 P.M. at the Waukesha County Technical Institute, New Berlin, WI. Call (414) 246-6634 for further details.

June 3 Southern Nevada Personal Computing Society will meet at Clark County Community College, Las Vegas, NV at 12:00. The club also meets on the 3rd Saturday of the month. For further information write SNPCS, 1405 Lucille St., Las Vegas, NV 89101 or call (702) 642-0212.

June 5 Minnesota Computer Society will meet at the Brown Institute, Room 51, 3123 E. Lake Street, Minneapolis, MN. For further information contact the Society at Box 35317, Minneapolis, MN 55435, Attn: Jean Rice.

June 6 Tidewater Computer Club will meet at the Electronics Computer Programming Institute, Janaf Office Bldg., Janaf Shopping Center in Norfolk. The club also meets on the 3rd Tuesday of the month. For details contact: C. Dawson Yeomans, Interface Chairman, 677 Lord Dunmore Dr., Virginia Beach, VA 23462.

June 7 New England Computer Society will meet in the cafeteria of the Mitre Corp., Rte. 62 in Bedford, MA at 7:00 P.M. Call Dave Day at (603) 434-4239 for details.

June 7 Kitchener Waterloo Microcomputer Club will meet at the University of Waterloo, Room 3388, Engineering Bldg. #4, University Ave., Waterloo, Ontario, Canada at 7:30 P.M.

June 7 The Valley Computer Club will meet at 7 P.M. at the Harvard School located at 3700 Coldwater Canyon, Studio City, CA.

June 7 Amateur Computer Society of Columbus will meet at the Center of Science and Industry at 7:30 P.M. For further information write c/o Fred Hatfield K8VDU, Computer Data Systems, 1372 Grandview Ave., Columbus, OH 43212, or call (614) 488-3347.

June 7 Lincoln Computer Club will hold its meeting at the South Branch Library located on 27th and South Sts. at 7 P.M. For more details write Hubert Paulson, Jr., 422 Dale Dr., Lincoln, NE 68510.

June 8 Mid America Computer Hobbyist meeting will be at 7:00 P.M. at Commercial Federal Savings & Loan, Bellevue NE. Intersection of Galvin Rd. and U.S. Hwy. 73-75. Write P.O. Box 13303, Omaha, NE 68113 for further information.

June 8 Utah Computer Association will meet at Murray High School, Rm 154, 5440 S. State St., Salt Lake City, UT at 7 P.M. For details write or call Larry or Holly Barney, 1928 S. 2600 E., Salt Lake City, UT 84108. (801) 485-3476.

June 8 The Rochester Area Microcomputer Society will meet at the RIT Campus, Rm. 1030, Bldg. 9 at 7:30 P.M. For details write RAMS, P.O. Box D, Rochester, NY 14609.

June 8 North Florida Computer Society will meet at 227 Edison Dr., Pensacola, FL 32505. For information write this address or call Eugene Rhodes at (904) 453-3844.

June 9 Northern New Jersey Amateur Computer Club (NNJACC) will hold its meeting at the Fairleigh Dickenson University, on the Rutherford Campus, Becton Hall, Room B8, at 7 P.M. For details write NNJACC, 593 New York Ave., Lyndhurst, NJ 07071.

June 9 HAUCC will meet at 7:30 PM in Rm 117 of the Science & Research Bldg. of the main campus of the Univ. of Houston. For more details write or call P.O. Box 37201, Houston, TX 77036, (713) 661-6806.

June 10 The Permian Basin Computer Group — Odessa Chapter meets at 1 P.M. in the Electronic Technology Bldg., Room 203 on the Odessa College campus. For details call (915) 332-9151.

June 11 North Orange County Computer Club will have its meeting at Chapman College, Orange, CA. Doors open at 12:00. 105 Hashinger Hall Auditorium. Membership Chairman, Tracey Lerocker, (714) 998-8080 evenings. For more information write P.O. Box 3603, Orange, CA 92655.

June 14 Home Computers Users Group for Radio Shack TRS-80 meets at 7:30 PM. For details write or call TRS-80 Users Group Information of Eastern Massachusetts, c/o Miller, 61 Lake Shore Road, Natick, MA 01760, (617) 653-6136.

June 14 Homebrew Computer Club will meet at 7 P.M. in Menlo Park, CA at the Stanford Linear Accelerator Center Auditorium. Call (415) 967-6754 for more details.

June 15 Northwest Computer Society meets in the Pacific Science Center in Seattle, Room 200 at 7:30 P.M. For more details write NCCN, Box 242, Renton, WA 98055.

June 16 Long Island Computer Association meets at 7 PM at the

Look To The North Star HORIZON Computer.

HORIZON™— a complete, high-performance microprocessor system with integrated floppy disk memory. HORIZON is attractive, professionally engineered, and ideal for business, educational and personal applications.

To begin programming in extended BASIC, merely add a CRT or hard-copy terminal. HORIZON-1 includes a Z80A processor, 16K RAM, minifloppy™ disk and 12-slot S-100 motherboard with serial terminal interface — all standard equipment.

WHAT ABOUT PERFORMANCE?

The Z80A processor operates at 4MHZ — double the power of the 8080. And our 16K RAM board lets the Z80A *execute at full speed.* HORIZON can load or save a 10K byte disk program in less than 2 seconds. Each diskette can store 90K bytes.

AND SOFTWARE, TOO

HORIZON includes the North Star Disk Operating System and full extended BASIC on diskette ready at power-on. Our BASIC, now in widespread use, has everything desired in a BASIC, including sequential and random disk files, formatted output, a powerful line editor, strings, machine language CALL and more.

EXPAND YOUR HORIZON

Also available—Hardware floating point board (FPB); additional 16K memory boards with parity option. Add a second disk drive and you have HORIZON-2. Economical serial and parallel I/O ports may be installed on the motherboard. Many widely available S-100 bus peripheral boards can be added to HORIZON.

QUALITY AT THE RIGHT PRICE

HORIZON processor board, RAM, FPB and MICRO DISK SYSTEM can be bought separately for either Z80 or 8080 S-100 bus systems.

HORIZON-1 $1599 kit; $1899 assembled.
HORIZON-2 $1999 kit; $2349 assembled.

16K RAM—$399 kit; $459 assembled; Parity option $39 kit; $59 assembled. FPB $259 kit; $359 assembled. Z80 board $199 kit; $259 assembled. Prices subject to change. HORIZON offered in choice of wood or blue metal cover at no extra charge.

Write for free color catalogue or visit your local computer store.

NORTH STAR ★ COMPUTERS
2547 Ninth Street • Berkeley, California 94710 • (415) 549-0858

CIRCLE INQUIRY NO. 40

New York Institute of Technology, Old Westbury Campus, Route 25A between Route 107 and Glen Cove Rd., Rm. 508. For more details write Long Island Computer Association, 36 Irene Lane East, Plainview, NY 11803.

June 16 Amateur Computer Group of New Jersey (ACGNJ) meets at UCTI, 1776 Raritan Rd., Scotch Plains, NJ 07076 at 7 P.M. For details write the above address.

June 17 San Diego Computer Society will meet at the Grossmont Community College Student Center, 8800 Grossmont College Dr., El Cajon, CA. Doors open at 12:30. For details call (714) 565-1738.

June 17 The 7C's Committee (Affiliated with the Cleveland Digital Group) will meet at Cleveland State University Student Services Bldg., in the Kiva Room at 2:00 P.M. For more information write to Cleveland Digital Group, 8700 Harvard Ave., Cleveland, OH 44105.

June 17 Philadelphia Area Computer Society will meet at 2 PM at LaSalle College Science Bldg. at the corner of 20th & Olney Ave. For more details write PACS, P.O. Box 1954, Philadelphia, PA 19105.

June 17 Computer Hobbyist Group of North Texas will meet at UTA University Hall, Rm 108 at 1 PM in Arlington, TX. For details call Neil Ferguson at (817) or (214) 265-9054.

June 17 Chicago Area Computer Hobbyist Exchange (CACHE) will meet at 1 PM in the Northern Illinois Gas Bldg., Golf and Sherman Rds., Glenview, IL. For details write CACHE, P.O. Box 52, So. Holland, IL 60473, or call CACHE Hotline, (312) 849-1132.

June 18 Central Florida Computer Club will meet at 2010 Fosgate Dr., Winter Park, FL 32789 2:00 PM. Contact Bill Kerns for details.

June 20 Rhode Island Computer Hobbyists (RICH) meets the at the Knight Campus of Rhode Island Junior College in the Faculty Cafeteria at 7:30 P.M. For details contact Emilio Iannucillo, RICH, P.O. Box 559, Bristol, RI 02809, or call (401) 253-5450.

June 23 Alamo Computer Enthusiast meets at 7:30 PM in Rm 104 at Chapman Graduate Center at Trinity University, San Antonio, TX. For details call (512) 532-2340, or write to the club at 7517 Jonquill, San Antonio, TX 78233.

June 25 Summit City Computer Club will meet at the McMillen Library on the Indiana Institute of Technology Campus in Ft. Wayne, IN. For details write the club at P.O. Box 5096, Ft. Wayne, IN 46805.

June 25 Birmingham Microprocessor Group will meet at Southcentral Bell Company headquarters bldg. at 2 P.M. For further details write or call Jim Anderson, 2931 Balmoral Rd., Birmingham, AL 35223; (205) 897-9630.

June 27 Sacramento Microcomputer Users Group, (SMUG), 7:30-9:30 P.M. at SMUD Training Bldg., on 59 St. Write Richard Lerseth, P.O. Box 161513 or call (916) 381-0335 after 5:00 P.M.

June 27 Computer Amateurs of So. Jersey will holds its meeting at the National Park Municipal Bldg., 7 So. Grove Ave., National Park, NJ at 7:30 P.M. For details call (609) 541-1010, or (609) 541-8296.

June 28 Diablo Professional Users Group (DPUG) will meet at Diablo Valley College Library, near the Willow Pass exit of Fwy. 680, from 8-10 PM. For details write or call Bob Hendrickson, Electronics Dept., DVC, Pleasant Hill, CA 94523; (415) 687-8373.

June 28 Ventura County Computer Society meets at Camarillo Public Library, 3100 Ponderosa Dr., Port Hueneme, CA 93041 at 7:30 P.M. For details write: VCCS, P.O. Box 525, Port Hueneme, CA 93041.

June 28 Boston Computer Society will meet at the Commonwealth School, 151 Commonwealth Ave., Boston at 7 P.M. The school is located on the corner of Dartmouth St. in Boston's Back Bay. For information write or call the society at 17 Chestnut St., Boston, MA 02108, (617) 227-1399.

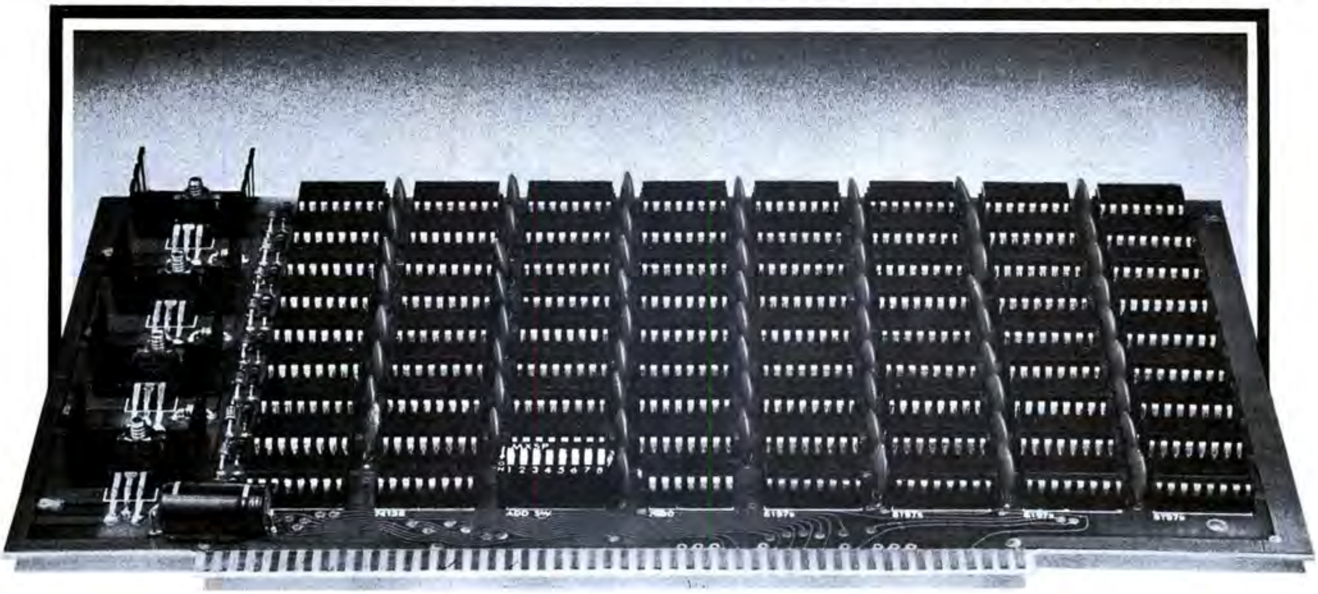
June 29 Space Coast Microcomputer Club will meet at 7:30 PM at the Merritt Island Library, Merritt Is., FL. Contact Ray Lockwood at (305) 452-2159 for details.

June 29 Small Computer Engineering Association of Minnesota (SCEAM) will meet at the Resource Access Center, 3010 Fourth Ave. So., Minneapolis, MN 55408 at 7 P.M. For more information write to this address or call (612) 824-6406.

June 30 Washington Amateur Computer Society has scheduled its meeting to be held at the Catholic University of America, St. Johns Hall. Located at Michigan and Harewood Aves. in Washington, D.C. Contact Bill Stewart at (202) 722-0210 for club details between the hours of 10 A.M. and 12 P.M.

SEALS is proud to announce...

OVER 20,000 UP AND RUNNING



The amazing SEALS 8K memory

April 1978 marked the second anniversary of the first Seals 8K memory. And over the past two years Seals has delivered more than 20,000 8K memories.

If you already own a Seals 8K board, you know about performance and dependability. But if you are still looking for a memory you can depend on, it's time you asked your local retailer about Seals reliability. Seals is the memory that works the first time, and every time.

The Seals 8KSC pictured above is available in both slow (500 nsec.) and fast (250 nsec.) versions. There are also Seals 8K's for your 6800 computer--the Seals 68KSC and 68KSC-Z. All Seals memory boards come assembled and tested and include gold edge connectors, sockets for all ICs, all address, control and data out lines fully buffered, and many more important features.

Pictured above is the first Seals 8K Memory Board # S/N 0001. The board that worked the first time.

When the time comes for your computer to remember more, you'll want to remember Seals--the memory people. Seals product reliability is a standard in the industry--the standard your system deserves. For performance and dependability you can count on, ask your retailer for memories and accessories from Seals Electronics.

The board pictured below is the very first Seals 8K ever assembled--S/N 0001. If you are a Seals 8K owner, you won't be surprised to know that this board has seen long and demanding service and was still up and running in a TVA computer in Tennessee when it was removed to be photographed.

Since Seals products are distributed through retailers throughout the country, we at Seals really don't know where all the 8K's are or how they are being used. If you already own a Seals 8K, we would like to hear from you. Just drop a note to our Marketing Division at 10728 Dutchtown Road, Concord, TN. 37922.

If you would like information on the full Seals product line, and a list of retailers in your area, write to the same address or call (615) 966-8771.

SEALS ELECTRONICS®
INCORPORATED

WHITE COLLAR MICROCOMPUTER

By James S. White

Don't buy a microcomputer — unless you, and your business, are ready for it. Today this column is the devil's advocate. In an issue of INTERFACE AGE filled with reasons to buy a microcomputer for your business, we are suggesting that there may be reasons not to. More precisely, there sometimes are compelling reasons not to buy some kinds of computers for some kinds of applications and environments.

"To everything there is a season, and a time to every purpose under heaven." There is, or will be, a time, and a computer, for every purpose, including yours.

You have heard of computer uses that have been extremely successful, and that have considerably improved the business fortunes of their users. Other combinations of computers and applications have been duds. Some have only been obstructions to the organization's getting its real work done. Others have sponged up considerable personnel time and other resources while getting very little in return. What kind of results you get, and the effect on your livelihood, is your decision as a business manager. But how to decide, to choose the computer meant for you, and the time to bring it into your organization?

DEFINE YOUR OBJECTIVES

First comes the decision of why you want a computer. This seems obvious, yet millions of dollars have been wasted on computers selected for truly unreasonable reasons. In many cases, these computers were procured for reasons totally unrelated to their planned use, or too soon, or too late to do the job for which they were intended. Prime examples are the status symbol computers some large businesses rushed to install during the middle 1960's.

If you don't know, to a fair degree of detail, what you will do with a computer, don't buy one. Wait until you know, and are fairly comfortable with your reasons, plan and schedule.

EVALUATE THE COMPLETE SYSTEM

The time for you to buy a computer is when you have found a complete computing service package that will meet your organization's objectives. Although most microcomputer systems offered today are appropriate for hobbyists, many lack features essential for the typical business user.

As a brief survey, the following elements are needed to complete the resource package necessary for most business computing applications:

Hardware — computing, data storage, input, and output equipment.

Maintenance — timely, competent service to keep the hardware operational.

Supplies — consumables well-suited to the hardware and application needs.

Software — programming to "train" the equipment to serve your unique organization and meet its application needs.

Applications planning — from a person knowledgeable of the computing system and of your type of business, to help integrate the two.

Many of today's vendors offer a few of these parts of a complete computer service package. Unfortunately, microcomputing products aren't highly standardized, because of the rapid growth of the field and its technology. In fact, few real standards exist. The user who procures various parts from different vendors is likely to end up with serious mismatches and overwhelming time costs for their integration — if it can be achieved at all.

MEDIUM RANGE PLANNING

Computer plans can easily be too short-sighted. In order to be of much value, a computer must become an integral, continuing part of a business. Therefore, the vendor supplying your computing service package should be one likely to be able to support you for the duration of your use of that computer.

Some users, when considering computers, forget that their businesses are likely to expand — to grow in volume, number of customers, types of products, etc. Most first-time computer users don't realize how much they will benefit from computers and that they are likely to later want their computer to do many more jobs than they first intended. Therefore an appropriate computer is expandable, able to grow with your business.

Computer plans can easily be too long-range to be valid or useful when considering long-range factors. Although planning for a second group of added computer uses is generally realistic, the first-time computer user can't know all the ways he will use a computer after a few years from now.

The microcomputing field itself is changing too rapidly to allow much realistic long-range planning. No one knows about some of the products that will be commonly available as soon as 5 years from now. Waiting is worse; the user who waits until something better isn't coming, will wait forever.

Even an organization's use of a computer shouldn't be expected to remain fixed for a long period of time. After the results of the first computing system use overhaul, or the second group of applications have been added, a total review is appropriate. Then it is generally worthwhile to seriously consider moving to an entirely new base of hardware and applications system design. Continued patchwork to meet changing needs over a period of years can result in an unresponsive, even unmanageable, computing system. Therefore, trying to plan in detail for a very long time of use of your first computer may be a worthless exercise.

For many first-time microcomputing users, a planning period of about 3 years is most appropriate. Equipment and your plans can be practically forecast for this long. For most small business applications, this amount of time will allow for initial applications to be fully implemented, and the completion of a full cycle of recognition of problems and implementation of solutions or improvements.

DON'T WAIT FOR PRICE CUTS

With the history of dramatic price drops in a variety of electronic products, many potential users are waiting for bargains, as compared to today's prices. However, the prices of complete business microcomputing service packages are unlikely to decrease much.

First, inflation is part of the world's economy, and will be for the foreseeable future. Many parts of a microcomputing installation are labor-intensive, and so can only

Branched to Page 51

Puzzled About Computers?

Data Dynamics Technology has a library of answers . . .

1 **Some Common BASIC Programs**
By Lon Poole & Mary Borchers
200 pages, $7.50
This book is designed for people who can use a variety of practical BASIC programs. The programs are written in a restricted version of standard BASIC and are compatible with many versions of BASIC now available to microcomputer users. Program descriptions, examples and remarks accompany the listings to help you follow BASIC programming logic.
Order No. OSB21002, paper.

2 **Payroll with Cost Accounting-In BASIC**
By Lon Poole & Mary Borchers
400 pages, $12.50
This book includes program listings with remarks, descriptions, discussion of the principles behind each program, file layouts, and a complete user's manual with step-by-step instructions, sample reports and CRT displays.
Order No. OSB22002, paper.

3 **Getting Involved with Your *Own* Computer: A Guide for Beginners**
By Leslie Solomon and Stanley Veit
216 pages, $5.95
This is the first major book by outstanding authorities that assumes only that you have an interest in what makes personal computers work and what can be done with them. It will not make you a computer expert, but it will teach you enough to understand what is going on and where to go for more information.
Order No. RID004-8, paper.

4 **Build Your Own Working Robot**
By David L. Heiserman
234 pages, $5.95
Here are complete instructions, plans, schematics, logic circuits, and wiring diagrams for building Buster, the most lovable (and mischievous) mechanical pet in the world! He'll serve you coffee or bring you the morning papers. He'll forage for his own "food" and scream when he can't find it. His "curiosity" will get him into one plight after another, but Buster has the capacity to get himself out of trouble just as easily as he got into it!
Order No. TB841, paper.

5 **Miniprocessors: From Calculators to Computers**
By David L. Heiserman. 195 pages, $5.95
This book will show you how to do it — in easy step-by-step stages — and at the same time bridge the gap between calculators and computers. Using readily available parts, you begin with a simple arithmetic calculator, then add basic unit, memory and programming functions. Eventually your project evolves into a functional hybrid calculator system with a fully programmable, randomly addressable 256-step memory, which "understands" machine language programming!
Order No. TB971, paper.

6 **Basic BASIC: An Introduction to Computer Programming in BASIC Language - Second Edition**
By James S. Coan. 288 pages, $8.95
The author uses over 100 sample programs to illustrate the essential techniques of the language and to integrate BASIC programming with mathematics. Each language statement or capability is clearly explained at the time that it is first used in a sample program. Every section is followed by practice problems.
Order No. HAY5106-9, paper.

7 **Game Playing with BASIC**
By Donald D. Spencer. 166 pages, $6.95
Enjoy the challenge of competition with your personal computer! Amuse yourself with such computer games and puzzles as 3-D Tic-Tac-Toe, Nim, Roulette, Slot Machines, Magic Squares, the 15 Puzzle, Go-Moko, Keno, Morra, Baccarat, Knight's Magic Tour, and many others.
Order No. HAY5109-3, paper.

8 **Microcomputers At A Glance**
By Donald D. Spencer. 192 pages, $7.95
This book contains about 2500 words, phrases, and acronyms used in conjunction with microcomputers. The keynote of this book is clarity — without sacrifice of authority and definitional precision. The book contains an introductory section which introduces the reader to microcomputers. The book is illustrated with diagrams and photographs. It is an essential tool for the increasingly large number of people involved in using microcomputers.
Order No. CAM021-8, paper.

9 **The Computer Quiz Book**
By Donald D. Spencer. 128 pages, $5.95
This book is written for readers who would like to test themselves on basic computer concepts. It may be used effectively by personal computer users, teachers, students, laymen, programmers, or anyone else interested in checking their knowledge of computer concepts. Exercises are grouped under the following headings: History, Applications, Equipment, Programming, Languages, Data Processing Concepts, Number Systems and Codes, and Occupations. The book contains several computer crossword puzzles and cartoons.
Order No. CAM025-0, paper.

10 **Your Home Computer**
By James White. 220 pages, $6.00
For the pre-hobbyist and the microcomputer novice, *Your Home Computer* provides a complete introduction to the world of home computing, beginning with what computers are and how they work. This book requires no prior knowledge or experience in electronics or computing. It provides answers to your many questions about hardware, software, and the personal computing scene today.
Order No. DMX 05-1, paper.

11 **Microprocessors: From Chips To Systems**
By Rodnay Zaks. 416 pages, $9.95
Microprocessors is a complete and detailed introduction to microprocessor and microcomputer systems. It presents both the concepts, and the actual techniques and components used to create systems. It introduces the reader to the aspects of system operation, use, and design.
Order No. C201, paper.

DATA DYNAMICS TECHNOLOGY P.O. Box 1217, Cerritos, CA 90701 5-78

Name (Print)__________
Address__________
City __________ State_____ Zip_____

Please send me:

Description	Qty	Price	Total	Book Order #	Qty	Price	Total
Binders		7.50					
Slip Cases		5.95					
Hex Code Cards		2.95					
Octal Code Cards		2.95					

Shipping & Handling Charges
Binders and Slip Cases $1.00 ea. U.S., $1.50 ea. Foreign
Code Cards $ .25 ea. U.S., $ .50 ea. Foreign
Books $ .50 ea. U.S., $1.00 ea. Foreign

TOTAL ORDER $_____
***TAX $**_____
SHIPPING & HANDLING $_____
TOTAL ENCLOSED $_____

☐ Visa #_____ ☐ M/C #_____ ☐ Check or M.O.(U.S. Funds drawn on U.S. bank)
Exp. Date_____ Signature_____

*California residents add 6% sales tax. Prices quoted subject to change without notice.
Please allow six weeks for delivery. You may photocopy this page if you wish to keep your INTERFACE AGE intact.
Orders *cannot* be shipped unless accompanied by payment, including shipping & handling and tax where applicable.
DATA DYNAMICS TECHNOLOGY, A Division of INTERFACE AGE Magazine (213) 926-9544

1

2

3

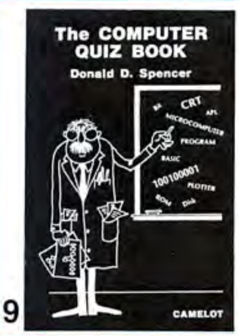
4

5

6

7

8



9

10

12 TYCHON's 8080 Octal and Hex Code Cards

The code cards are a sliderule-like aid for programming and debugging 8080 software. Both cards contain all the standard mnemonics and either their corresponding octal or hex codes. The pocket size cards are 6.5 by 3 inches (16 by 8 cm) with color-coded instructions to provide a neat, logical format for quick reference. The back of both cards is printed with an ASCII code chart for all 128 characters plus the 8080's status word and register pair codes.

13 INTERFACE AGE Binders and Slip Cases

Collecting magazines can bring headaches — not to mention dust, ripped pages and misplaced copies. If you use your back issues of INTERFACE AGE as reference material, nothing is more annoying than taking time to find mislaid copies. Data Dynamics Technology is now offering deluxe binders and slip cases which will place each back issue of INTERFACE AGE at your fingertips. Each binder and slip case is constructed of a handsome blue vinyl with INTERFACE AGE stamped in gold foil on the front cover and spine. These rugged binders and slip cases can hold 12 issues each and will protect your back issues of INTERFACE AGE for years.

Robots On Your Doorstep
By Nels Winkless & Iben Browning
175 pages, $6.95
Order No. DP12-6, paper.

8080 Machine Language Programming for Beginners
By Ron Santore. 104 pages, $6.95
Order No. DP14-2, paper.

Home Computers: A Beginners Glossary and Guide
By Merl K. Miller and Charles J. Sippl
147 pages, $6.95
Order No. DP02-9, paper.

Home Computers: 2^{10} Questions & Answers
Volume 1: Hardware
By Rich Didday. 225 pages, $7.95
Order No. DP00-2, paper.

Home Computers: 2^{10} Questions & Answers
Volume 2: Software
By Rich Didday. 175 pages, $6.95
Order No. DP01-1, paper.

A Step by Step Introduction to 8080 Microprocessor Systems
By David L. Cohn & James L. Melsa
169 pages, $7.95
Order No. DP04-5, paper.

Computers & Their Uses
By W.H. Desmond. 427 pages, $13.50
Order No. PH5498-5, cloth.

Advanced BASIC
By James S. Coan. 192 pages, $7.95
Order No. HAY5855-1, paper.

An Introduction to Microcomputers: Volume 1, Basic Concepts
By Adam Osborne. 282 pages, $7.50
Order No. OSB2001, paper.

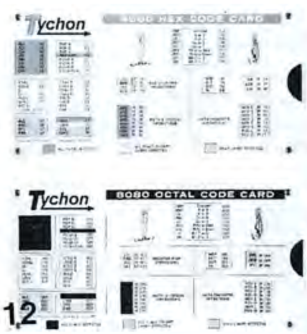
APL: A Short Course
By Sandra Pakin and the Staff of Computer Innovations. 157 pages, $10.50
Order No. PH8877-7, paper.

Digital Trouble Shooting: Practical Digital Theory & Troubleshooting Tips
By Richard E. Gasperini. 180 pages, $9.95
Order No. HAY5708-3, paper.

Digital Experiments: Workbook of IC Experiments
By Richard E. Gasperini. 177 pages, $8.95
Order No. HAY5713-X, paper.

Prices quoted subject to change without notice.

13



12

11

THE MIND REVOLUTION

By Merl Miller

What is a robot? What is intelligence? These questions have caused a lot of speculation, concern and conversation. We might say that intelligence is the ability to think, learn and solve new problems. We could then say that a robot is a *machine* capable of carrying out these functions on its own. This may not be the best definition in the world, but it is a workable one. It may be easier to discuss what a robot is *not* than what a robot *is*.

Let's start by discussing controllers. A controller is a programmable electronic machine designed to do a specific job. Consider one device inside this machine. For an input of x it has an output of y. It has a gain of G that relates to y and x in this manner: $x = Gy$

The input to a controller is a signal of a frequency, f. When we input a signal of x, at a certain frequency, we can measure the output at y. We can then compute the gain for our machine by dividing y by x:

$$G = \frac{y}{x}$$

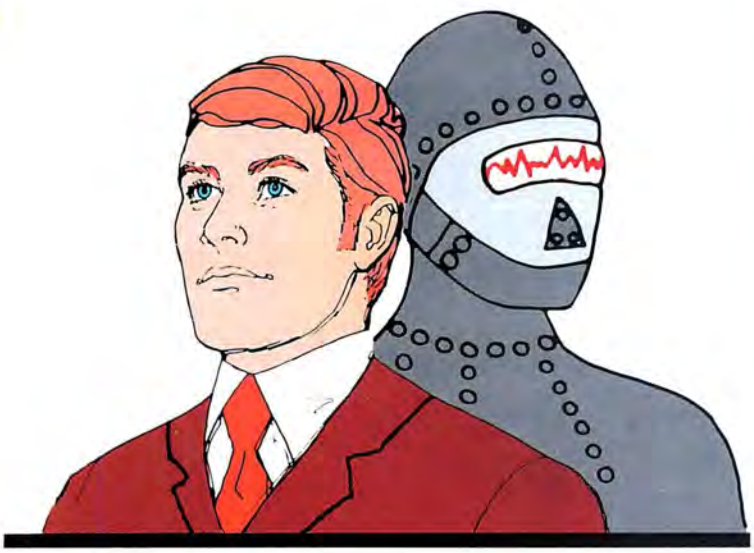
By repeating this step for different frequencies we can determine a gain characteristic. (We'll come back to this later.) It can be shown mathematically that when the gain characteristic is a constant, the output y is almost identical to the input x.

. . .a robot is a machine capable of carrying out these functions on its own. This may not be the best definition in the world, but it is a workable one.

There will be a group of frequencies which provide this constant gain. This is referred to as the device's operational bandwidth.

Now, what if we put a large group of these devices into a machine? For simplicity's sake, let's assume we can construct the electromechanical apparatus necessary to make our machine work. Let's also assume that each of our devices works on a different frequency and we've been smart enough to use a large variety of frequencies so we don't confuse our devices. We will put our machine into a mannequin and have separate devices to control the head, arms, legs and torso.

If we program a microprocessor to send various signals to our machine, we can let it run by itself. We can then determine the length of time it will run. It will walk, wave its arms, salute, bend in the middle and do a lot of other "magic" things.



Our limitations are determined by what kind of programmers we are and what kind of machine we build. Everyone will be amazed at our marvelous robot! There is only one little problem, however; we have built a programmable controller, not a robot.

If you accept my definition that a robot is a device capable of carrying out intelligent functions on its own, then you have to accept the idea that a programmable controller is not a robot. A robot, like a human, must function as an interactive system; so to understand the concept of artificial intelligence, we must start with some elementary concepts of systems theory. There is a little math involved in this, but please don't let that scare you off.

Let's go back to gain characteristic. We have already determined that if input a signal of x into a device, the result is y. In a controller (or a robot, for that matter), we may feed in an input of x at a frequency f, measure its output y and compute its gain ratio,

$$G = \frac{y}{x},$$

at this frequency. By repeating this step for different frequencies, we obtain values of G at different frequencies. We can plot a gain characteristic of G, as shown in Figure 1. When the gain characteristic is constant, that is between f_a and f_b, y is almost identical to x. The bandwidth then is the area between f_a and f_b.

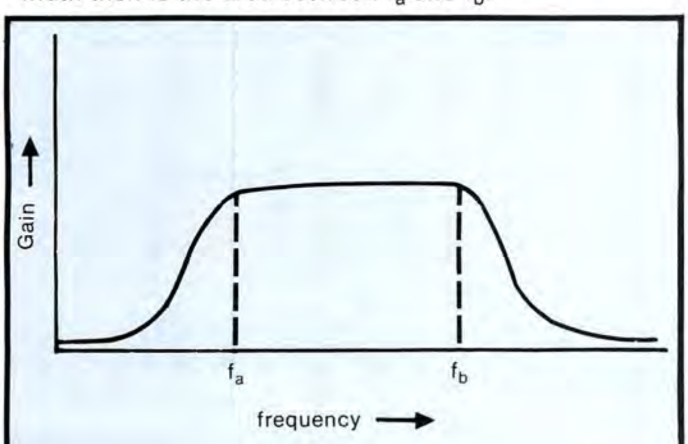
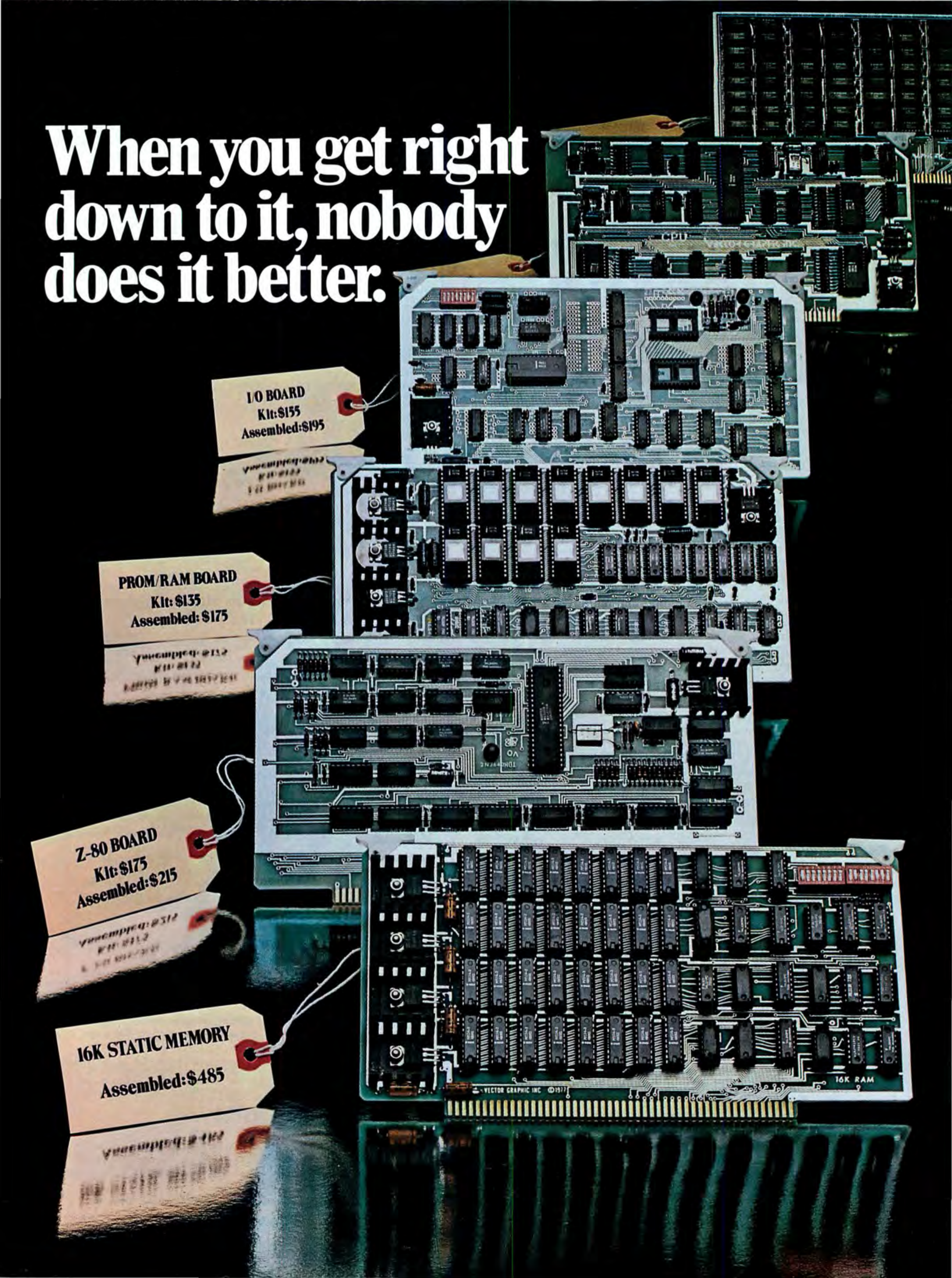


Figure 1.

Part of learning and, thence, part of intelligence, is the ability to do something repetitively and modify your behavior as you go. We can best examine how a robot might do this by looking at a system model of a robot.

Branched to Page 36


When you get right down to it, nobody does it better.
I/O BOARD
Kit: $155
Assembled: $195
PROM/RAM BOARD
Kit: $135
Assembled: $175
Z-80 BOARD
Kit: $175
Assembled: $215
16K STATIC MEMORY
Assembled: $485
VECTOR GRAPHIC INC ©1977
16K RAM

"12 persuasive reasons why Vector Graphic gives you more microcomputer value for your money."

Bob Harp, Technical Director

1 Our products are designed by professionals with many years of experience in industrial electronics.

2 Only prime components are used from vendors you will recognize as the most reputable in the industry.

3 Our kits are designed to be easily assembled with a minimum of user installed jumpers — you don't have to spend hours figuring how to configure boards for your application.

4 Our cabinets quality is superior — designed with careful attention to airflow (a fan is standard) and mechanical rigidity.

5 We offer a complete line of products designed to operate together — no need to be concerned about compatibility.

6 You have a choice of CPU boards with either 8080 or Z-80 MPU.

7 We offer all of our products assembled and tested — for surprisingly little more than kit prices.

8 Our management is genuinely interested in you as a satisfied customer.

9 You can purchase our products with confidence — they will not become obsolete with the next generation of microprocessors.

10 We offer complete technical support with engineers and technicians available to answer your questions.

11 Our speedy shipment of orders is the envy of our competitors.

12 A network of over 150 dealers throughout the U.S., Canada and Europe has been carefully selected to service your needs.

No other company in the industry has grown as rapidly as Vector Graphic.
If you want to know why Vector Graphic has outstanding acceptance, purchase one of our products.
Write or call for your nearest dealer.

© VECTOR GRAPHIC INC.

Vector Graphic, Inc., 790 Hampshire Rd., A + B
Westlake Village, CA 91361, Tel: (805) 497-6853

Vectored from Page 33

Device A (Figure 2) is a sensing device. It takes two inputs, x and z, and output w, which is the difference between x and z. The result is then read at y. Next, y is fed back to the device as z and the process is repeated. This is a simple feedback control loop.

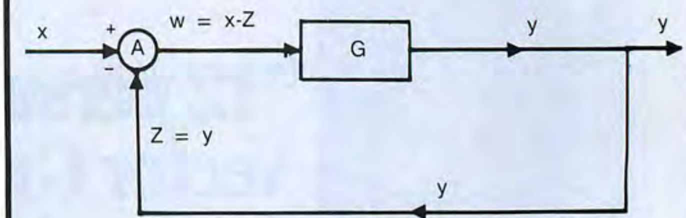


Figure 2.

Let's add an additional gain, H (Figure 3). We now have the following relationships:

$$z = Hy \qquad w = x\text{-}z \qquad y = Gw$$

These are simultaneous equations so we can eliminate the variables x and z. This is done in the following manner:

1. $w = x - Hy$
2. $y = G(x - Hy)$

therefore $y - Gx - GHy$

3. $Gx = y + GHy$
4. $Gx = y(1 + GH)$

or $y - \dfrac{G}{1+GH}x$

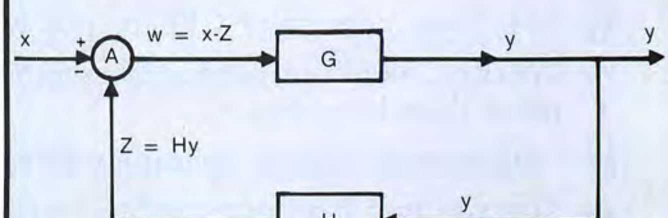


Figure 3.

This, then, is the fundamental equation for a robot. If we define system gain as

$$sG = \frac{y}{x}, \text{ then:}$$

$$sG = \frac{G}{1+GH}$$

Remember, gain is always equal to output divided by input.

System gain is such an interesting subject that large portions of books have been written about it. We will get into how it works and how it can be utilized by a robot next month. □

JURISPRUDENT COMPUTERIST

By Elliott MacLennan
Attorney-at-Law
Stephen Murtha

SUBCHAPTER C CORPORATIONS
Tax Slasher and Workhorse, Part II

Last month we discussed the rudimentary legal and tax structure for the Subchapter C corporation. In this article, we will discuss some of the tactical tax footwork that makes this "workhorse" of the various corporate entities the unique form of business that it is.

CANDIDACY FOR SUBCHAPTER C:

The Subchapter C corporation is taxed at a uniform rate with a maximum corporate tax of 48 percent. Individuals, sole proprietors, partners, co-owners, and Subchapter S shareholders are taxed at the graduated rate of 14 percent to 70 percent. Whether or not your business is a candidate for a Subchapter C corporation; that is to say, is it profitable to become a Sub C, in that your business will actually save hard tax dollars, cannot be determined by a formula or by a stated dollar amount of income per principal in the corporation. If your business has a gross income of $200,000 in one year, and it has expenses including salaries of $200,000, incorporation into a Subchapter C will not help tax wise. The same holds true were the figures $10,000,000.

Whereas, if your firm's gross income is $200,000 and it spends $160,000, a tax savings will result. Why the difference? Where after the payment of business expenses, including salaries, a profit is left over (retained earnings), and where the owners (shareholders) of the business are in a higher personal tax bracket on the graduated scale than the corporation is on its uniform scale, then a tax savings results.

Tax Saving Example:

For example, if the sole owner of a Subchapter C corporation (it only takes one person to incorporate in California) is in a personal tax bracket of 50 percent and his corporation, after paying his salary, also pays out to him the $40,000 in profits, he will be taxed a minimum of $20,000 on that $40,000, and a maximum of $28,000 (70 percent maximum tax on dividend distribution from a corporation), $20,000 or $12,000 will be left.

If the owner retains this profit in the corporation, he will pay $8,300 for a substantial net hard tax dollar saving as compared with paying the profit out as a dividend noted above.

The fly in the ointment is that if the owner leaves this profit in the corporation, he cannot have it at his personal disposal. The $40,000 in profit is now $31,700. The owner can use this profit to make a down payment on the purchase of a business asset. If he does not have the corporation purchase the asset, his down payment would only be $20,000 or, worse yet, $12,000. Plus, more of each dollar paid by the corporation goes toward paying off the newly acquired asset, rather than going into the U.S. Treasury, than if the owner had bought the asset in any form other than a Subchapter C corporation.

Additionally, the depreciation and investment tax credit will serve to reduce corporate income even more, making less tax to pay. Plus, the owner can elect an accelerated depreciation schedule to get his write-off in the early years of the useful life of the asset. Caution: Although accelerated depreciation can greatly enhance quick write-off of an asset, it is not without pitfalls; therefore, the reader, like the hypothetical owner, should seek competent tax advice before so electing depreciation acceleration.

Moreover, since this asset is a new purchase, and if it has a useful life of six years or more, the owner can take an additional special "first-year depreciation allowance" of up to $2,000, in addition to the investment tax credits, and straight-line or accelerated depreciation. Caution: Do not buy the asset on which you plan to take first-year depreciation from a blood relative.

It may seem a bit unfair to you, the reader, to state succinctly that the chief reason why the Sub C saves tax dollars on the basis of this one example, as opposed to providing you several examples and let you derive the simple answer yourself, but since we are talking about dollars and not puzzles, I will state the reason: The Sub C is a timing device. It allows its shareholder-owners to time the receipt of personal income to themselves in a way no other business form can do. Again, where the corporation is in a lower bracket than the shareholder-owner, it is advantageous tax wise to retain the profits and purchase business assets where the desire is to acquire more equipment, expand, or to time the distribution of profit when the owner is in a lower personal tax bracket; or to invest the profit, or a portion of it, into one of the unique pension and profit sharing plans available only to a corporation.

The superimposition of other tax election planning devices, straight-line or accelerated depreciation for example, on top of the basic Sub C timing device, presents a synergistic approach to lowering taxes.

The Sub C can also be used as an excellent income splitting device to lower taxes without being subject to Internal Revenue Service's attack on grounds of an anticipatory assignment of income. It works this way: Assume a sole proprietor or a partner decides to "gift" some of his income to his two children because he desires to get some of his income out of his 35 percent bracket into his children's bracket of 14 percent, thus lowering his overall tax burden while directing the income (and tax savings) into the same place it went before; his family. The IRS will call this a prohibited anticipatory assignment of income for tax avoidance purposes, and a "tax the tree that produced the fruit," under a special section in the Internal Revenue Code known as the "silent policeman." The sole proprietor or partner will definitely pay all the income tax on this aborted income-split, plus, he may pay a gift tax to boot.

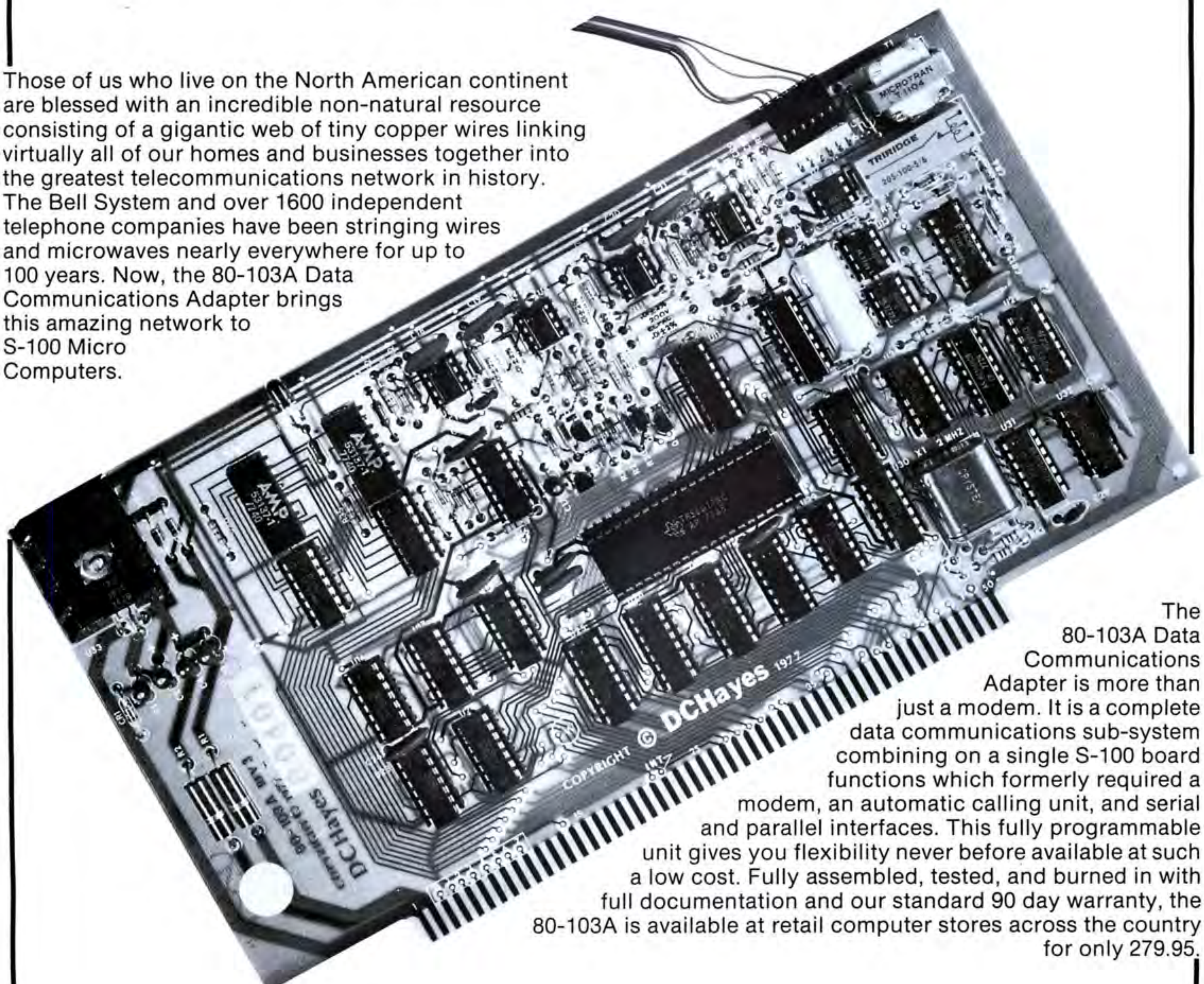
If the income "tree" is a shareholder owning stock, then the gift of stock and the dividends paid thereunder is not deemed to be an assignment of income, thus achieving the desired income split. Another tax trick is to gift the stock before substantial capital appreciation has occurred with respect to the stock, so that the shareholder and his spouse can gift up to $6,000 for each child, each year, without payment of gift tax. This is known in the tax trade as "the problem you hope gets worse." Another effect of the gift of stock that rapidly appreciates is that it removes the substantial appreciation from estate and inheritance taxes imposed upon the death of a shareholder or his spouse. And lastly on this subject, either gift non-voting stock to the children, or voting stock held in a voting pool, or a trust arrangement with you-know-who in control of how the vote is cast.

In the next article, we will explore the tax fringe benefits available to a Subchapter C corporation, and also mitigating the effect of business failure by a unique corporate tax device.

The Subchapter C corporation can be used, because of its unique timing features, as an income averaging device to filter out wild fluctuations in income (income "spikes"). The standard "income averaging" provisions can only filter out little ripples. The effect is to flatten out income where it is distributed to shareholder-owners, to avoid the graduated, regressive 14 percent to 70 percent income tax scale. □

modem / 'mo • dəm / [**mod**ulator + **dem**odulator] *n* - *s* : a device for transmission of digital information via an analog channel such as a telephone circuit.

Those of us who live on the North American continent are blessed with an incredible non-natural resource consisting of a gigantic web of tiny copper wires linking virtually all of our homes and businesses together into the greatest telecommunications network in history. The Bell System and over 1600 independent telephone companies have been stringing wires and microwaves nearly everywhere for up to 100 years. Now, the 80-103A Data Communications Adapter brings this amazing network to S-100 Micro Computers.



The 80-103A Data Communications Adapter is more than just a modem. It is a complete data communications sub-system combining on a single S-100 board functions which formerly required a modem, an automatic calling unit, and serial and parallel interfaces. This fully programmable unit gives you flexibility never before available at such a low cost. Fully assembled, tested, and burned in with full documentation and our standard 90 day warranty, the 80-103A is available at retail computer stores across the country for only 279.95.

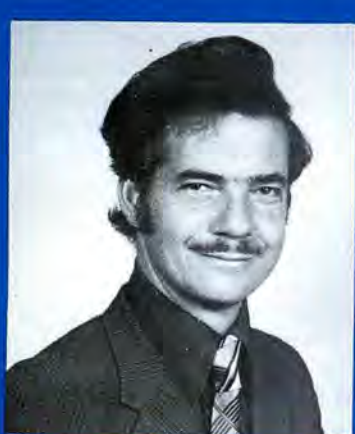
D.C. Hayes Associates Inc.

P.O. BOX 9884 • ATLANTA, GA. 30319 • (404) 231-0574

Distributed in Canada by TRINTRONICS LIMITED, Toronto

... FROM THE FOUNTAINHEAD

By Adam Osborne



It is time for another product update, since there has been an avalanche of new product announcements from microprocessor manufacturers.

Intel has now made public some details of the 8086; the most interesting aspect of this microprocessor, from the hobbyist's viewpoint, is the fact that it will execute 8080A programs without modification. This will make the 8086 far more interesting to the hobby market, since it does not require software to be regenerated from scratch. The 8086 is supposed to have a computing capacity that is ten times that of the 8080A; however, a great deal of this extra computing capacity derives from its enhanced instruction set. If you are going to execute existing 8080A programs on the 8086, you may double your throughput — but that is not to be sneezed at. However attractive all of this sounds, there will be no 8086's on the market for at least another six months, which means that you will not see any available for hobbyists' use until the middle of 1979. Nevertheless, you should keep a sharp eye out for the 8086 developments, since the 8086 will let you use what you now have (providing you have an 8080A-based system) while simultaneously developing new products for the future.

Fairchild is now manufacturing 9440 Central Processing Units with excellent yields. The 9440 is a microprocessor that executes the Data General Nova 1200 instruction set. Fairchild will probably have S100-compatible 9440-based CPU and memory combinations available by the middle of this year. For those of you who need a lot more Central Processing Unit performance, this may be the way to go. But one word of caution: Data General, which manufactures the Nova line of minicomputers, is very unhappy with Fairchild for building the 9440. Data General sees the 9440 cutting heavily into the Data General customer base, doing so by executing programs that run on any Nova 1200 minicomputer. Data General themselves wrote at least part of the software running on most Nova 1200 minicomputers, and they do not want Data General's own software efforts to help Fairchild take away Data General markets. Data General can be expected to look very closely at all software being run on 9440-based microcomputer systems. Any software that can be shown to come from Data General itself is likely to have significant legal consequences. So be careful.

Data General's own MicroNova system has an instruction set which differs in small ways from other Nova instruction sets. Data General has been marketing its MicroNova system through selected stores, but the chip itself has not been advertised or promoted widely.

We still do not know when Zilog will have the Z8000 on the market. No announcements have been made about this product, so we must discount it, at least in the personal computing market, for the rest of 1978.

It would appear as though Texas Instruments is going to make its big push during 1978 with the 9440. This is a one-chip version of the TMS-9900, containing 2048 bytes of erasable, programmable read-only memory for data storage. This product is not particularly interesting to personal computing; it is an industrial product. There are now a number of support devices available for the TMS9900, including the TMS9901 programmable system interface, the TMS9902 asynchronous communications controller, the TMS9903 synchronous communications controller and, soon to come, the TMS9901 direct memory access controller. I do not expect Texas Instruments itself to make a significant impact in the personal computing market with a TMS9900. They have frittered away their chances for too long. Fortunately, companies like Technico are still around. But those of you who need a 16-bit microprocessor had better give Technico a lot of support. Texas Instruments' remarkably low profile has not helped the TMS9900 pioneers one bit.

For those of you who are doing a lot of scientific work, the AMD9511 arithmetic processor is now finally available, but it is not cheap. In small quantities, I believe this part still costs more than $200.00 apiece. However, for transcendental functions, there is nothing on the market like it. It brings floating point arithmetic, logarithms, and trigonometric functions into the approximate price range of personal computing.

Now, there is one interesting development which may have a very significant impact on personal computing during 1979. We have all seen the time, expense and agony associated with taking a chip such as the 8080A and converting it into a working microcomputer system with adequate peripherals and software. The development of adequate software, in particular, has been so painful and slow that you will be excused for believing the 8080A and 6800 are going to be around until the year 2000, simply because no one has the fortitude to go through new software development for a new micro-

Introducing the SPACE BYTE™ MODULAR BUSINESS COMPUTER

Complete with assembly language application software

Complete as pictured $5,900.

- SB85-16 terminal mounted mainframe with the SPACE BYTE 8085 self contained computer and 16K SPACE BYTE fully static RAM (48K RAM capacity)
- HAZELTINE 1500 video display terminal
- Dual iCOM flexible disk drive system
- Floor stand
- SPACE BYTE "BIZPAK" assembly language disk interactive business application software package optimized by the speed of the 8085 CPU (requires only 16K RAM)
- iCOM FDOS III operating system

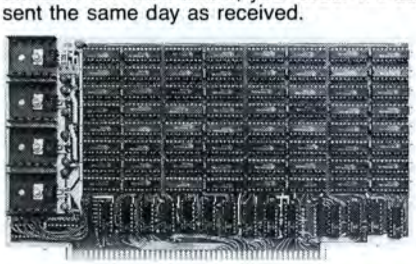
THE SPACE BYTE COMPUTER CORPORATION
1720 Pontius Ave., Suite 201
Los Angeles, CA 90025 (213) 468-8080

Now, Artec has an expandable elephant for everyone!

8K-32K of static RAM memory. Fully assembled or in kit form.

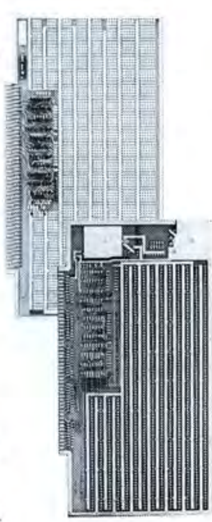
No matter what your needs, Artec has a memory board for you. You can start with 8K of TI 4044 memory on a 5.3" x 10" card and work your way up to a full 32K in 8K increments. The access time is only 250ns. The memory is addressable in 4K blocks and is perfect for S100 and battery augmented systems. The Artec 32K Expandable Memory has four regulator positions, bank select and plenty of room for all necessary support hardware. It uses less than 1 amp per 8K of memory (3.9 for 32K), and only +8 volts.

For five years Artec craftsmanship and reliability has been proven in tough industrial use. Now, you too can enjoy breadboards and memories that will work time after time. Send for an Artec board, your order will be sent the same day as received.



Kit: Board &	Fully Assembled Board:
8K of memory—$285.00	8K—$310.00
8K add on —$250.00	16K—$545.00
Full 32K board—$935.00	24K—$790.00
	32K—$985.00

GP100—$20.00
Maximum design versatility along with standard address decoding and buffering for S100 systems. Room for 32 uncommitted 16 pin IC's, 5 bus buffer & decoding chips, 1 DIP address select switch, a 5 volt regulator and more. High quality FR4 epoxy. All holes plated through. Reflowed solder circuitry.



WW100—$20.00
A wire wrap breadboard, similar to the GP100. Allows wire wrap of all sizes of sockets in any sizes of sockets in any combination. An extra regulator position for multiple voltage applications. Contact finger pads arranged for easy pin insertion.

Buffering Kit—$12.65
All the necessary components to bootstrap any Artec board into your system. Buffering I/O, DIP switch heat sinks and every support chip you need.

TO ORDER: Use your Mastercharge or BankAmericard. Or just send along a money order. We can accept only U.S. currency. Please include $3 handling on all orders. California residents add 6.0% sales tax.

FOR MORE INFORMATION: For more information about these or any of Artec's complete line of circuit boards or for either industrial or personal use, please call or write. A catalog will gladly be sent.

Please send me: (Include Quantity)
____ 32K ____ GP100 ____WW100
☐ I've enclosed a money order.
☐ Mastercharge No. ________ ______ Exp. Date
☐ BankAmericard No. ________ ______ Exp. Date
Name ______________________
Address ______________________
City ____________ State ______ Zip ______
Calif. Res. add 6% Sales Tax $3.00 Handling Encl.

10% discount for students & computer club members.

ARTEC ELECTRONICS, INC.
605 Old County Rd., San Carlos, CA 94070
(415) 592-2740

CIRCLE INQUIRY NO. 3

processor. A British company may just change all of that. The company is called CAP-CPP, Inc. Their American representative is John Payne. His address is: CAP-CPP, Inc.
299 California Street
Palo Alto, CA 94302

What CAP has done is develop a complete floppy disk operating system which supports a subset of COBOL for business applications. Now, this in itself is no big deal; but, they have gone a step further and designed the system to be machine independent. That is to say, they only need to write a few very low-level system programs in the language of a new microprocessor in order to convert the entire system so that it will run on the new microprocessor. While we have heard claims similar to this one before, we will watch with interest what CAP actually produces. Clearly, if they can come up with a higher level language and operating system that can be converted quickly and inexpensively to run on any new microprocessor, then these guys are sitting on a gold mine. And they may completely transform the economics of new products entering the personal computing marketplace.

Bryce Ward, who sold various board products under the name "Associated Electronics," recently went out of business. I received numerous complaints regarding Associated Electronics products that did not work, or were not delivered. I did not publish the name of Bryce Ward or Associated Electronics, and I believe I was right in my decision. Bryce Ward is not a crook, he is a very capable and honest man who was over his head in financial matters. As far as I am concerned, the people who ordered products from Associated Electronics, paying cash up-front are just as responsible for the failure of Associated Electronics as Bryce Ward, who went into the venture under-capitalized. My hope is that Associated Electronics customers stay active in the market — paying for goods after receiving them. I hope that Bryce Ward stays active designing hardware — in a fiscally sound environment.

I received a very important telephone call from Bill Burton, a New York area hobbyist. I mentioned the Structured Systems Group, an Oakland software company, in February. The Structured Systems Group provides applications programs written in C-BASIC. Mr. Burton ordered a program from Structured Systems Group, but received an object program only. The program worked as advertised, but Mr. Burton was unhappy because he expected to receive the source program, which he could modify as needed. Structured Systems Group stated that the source program would cost an additional $100.

In this matter, I believe Structured Systems Group is right and Mr. Burton (probably together with many other hobbyists) do not understand normal software sales practices. Unless a company specifically advertises that source programs will be provided, you must expect to receive object code only. This is standard practice in the software industry.

Software houses are reluctant to sell source programs, because this is their only protection against software theft. It is, of course, possible to recreate the source programs from object code, but the task is hard enough to discourage casual thieves.

I asked Mr. Burton whether he had requested his money back from Structured Systems Group. He said, "No, he needed the programs, it was just that he has anticipated getting the source codes." It is unreasonable to expect that any company will modify its policies based on unfounded customer expectations. I hope that in a situation like this, Structured Systems Group would return Mr. Burton's money, if he asked for it, but as matters stand, I cannot fault the Structured Systems Group.☐

EUROPEAN

INTERFACE

By Hans Drewitz and Roland Hesse

ANOTHER EUROPEAN COMPUTER STORE

What do you call a computer shop in France? COMPUTER BOUTIQUE™! Just a few minutes from the famous "Place de l'Etoile" in Paris, it offers a wide range of products covering the various needs of a large, but yet undeveloped market. The products are generally imported, mainly from the United States. "Computer Boutique has from the beginning applied a quality policy that has led to the development of a line of systems assembled in France from various imported components," said Manager Monsieur Peuplas, when we met him in his shop. "Among the most difficult problems to be solved during the first months by the management was the choice of reliable suppliers and equipment. Being so far away from Silicon Valley makes it a challenge, even with several visits to key suppliers, to provide the kind of services that Computer Boutique offers its customers: Fast delivery, expert maintenance and consistency in the products specifications," he continued. Computer Boutique delivers and maintains the following systems:

- NSC SC/MP, Motorola MEK D2, Dolphin (a Swiss-made system for beginners with a choice of several microprocessors);
- CB 6800: the full line of SWTPC products, including printer and disk systems;
- CB 100: an S100, Z80 system plus 8K memory, keyboard, TV display, cassette interface, all under one cover;
- CB 7700: a continuation of CB 100 products. Up to 64K and dual diskette (PerSci or DRI);
- CB 7716: the Alpha Microsystem.

Computer Boutique offers a full range of services, from training classes to financing, and also maintains contacts with software houses and consultants for the design and programming of dedicated applications. To be a professional organization, in a field that started by attracting amateurs, is the objective of Computer Boutique.

With this experienced team, and the backing of Sigmatronics — a privately-held French company, distributors of

INFO 2000 DISK SYSTEMS
NEW! Let You Make the Choice

INTELLIGENT CONTROLLER or DUMB CONTROLLER

INFO 2000 has just added a new, lower cost S-100 "dumb" controller disk system to its already popular line of "intelligent controller disk systems for S-100, Digital Group, and Heathkit H8 microcomputers.

All INFO 2000 disk systems feature the incomparable PerSci 277 dual diskette drives with voice-coil positioning and seek time up to eight times faster than competitive stepping-motor-type drives. All INFO 2000 disk systems are fully assembled, rigorously tested, and include case, power supply, cables, and software. All INFO 2000 disk systems include the CP/M Disk Operating System and support the complete INFO 2000 library of software for 8080, 8085, and Z80-based systems. All INFO 2000 disk systems include complete I/O driver software *CUSTOMIZED* for your specific equipment configuration at no additional cost, assuring you of a hassle-free plug-in-and-go installation.

And now, S-100 users can choose the kind of controller that makes the most sense for their specific application: intelligent or dumb.

INTELLIGENT CONTROLLER DISK SYSTEM

This system uses the remarkable PerSci 1070 "intelligent" controller, which incorporates its own dedicated 8080 microprocessor, 4K of EPROM containing extensive file management firmware, 1K of RAM buffer memory, and eight-bit parallel interface.

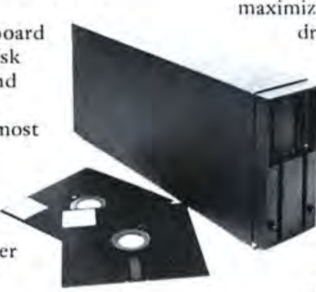
The "intelligent" controller is actually a single-board computer dedicated to the task of managing the disk system. It requires a minimum of interface logic and very little support software in the host computer. This makes it exceptionally easy to interface to almost any kind of computer system or software system.

INFO 2000 provides interfacing hardware (Adapter Boards) for all S-100, Digital Group and Heathkit H8 microcomputers. The Adapter Boards provide all necessary interfacing logic, power regulation, and support an EPROM-resident CP/M bootstrap loader. The Heathkit H8 Adapter Board replaces the Heath 8080 CPU board and upgrades the H8 to a Z80.

If you change to a different kind of computer in the future, you can still use your disk drive and controller. You need only purchase the appropriate replacement INFO 2000 Adapter Board.

Prices for the **complete INFO 2000 Disk System** with "intelligent" controller:

$2,850 for S-100 or Digital Group
$2,950 for Heathkit H8



DUMB CONTROLLER DISK SYSTEM

This new system uses the new S-100 controller board developed by INFO 2000 Corporation especially for our own Business System. The new "dumb" controller is substantially less expensive, and is designed specifically to maximize the performance of the PerSci 277 dual diskette drive when used in a CP/M software environment.

The controller is fully IBM 3740 compatible, and provides advanced functions not often found in low-cost units: full soft-sectored diskette formatting, multi-sector reads and writes, verified seeks, and complete diagnostic capabilities.

This new controller is FAST! A full verified disk copy takes less than a minute. Formatting a new diskette takes less than half a minute. A CP/M re-boot is almost instantaneous (one-third of a second). There are no performance compromises.

The INFO 2000 controller is available with an "I/O Option". This adds two RS232 serial ports with software-selectable baud rates, 3 8-bit parallel ports (2 output, 1 input), and sockets for an additional 7K of 2708-type EPROM (1K is standard). All of this is contained on the same S-100 board which holds the controller, and the cost of the "I/O option" is $150— far less than the cost of a comparable serial/parallel interface board and an EPROM board.

$2,450 for INFO 2000 S-100 DISK SYSTEM

SUPPORT SOFTWARE

INFO 2000 Disk Systems are supported by the most extensive library of software available anywhere. All INFO 2000 Disk System prices include the Digital Research CP/M Disk Operating System and an EPROM containing I/O driver software customized for your specific hardware configuration. INFO 2000 also offers a choice of three BASICs, two FORTRANs, three assemblers, two text editors, a word processing package, a fast sort package, and much more software. Write or phone to receive our brochure with full details.

You may also be interested in the INFO 2000 Business System—a complete data processing system for small businesses, with full accounting and word processing software, and priced under ten thousand dollars.

INFO 2000 CORPORATION
20630 South Leapwood Avenue
Carson, California 90746
(213) 532-1702

CIRCLE INQUIRY NO. 30

INFO 2000 DISK SYSTEMS

microcomputer equipment—Computer Boutique will find its place in the market. Bonne chance, Computer Boutique!

WHAT DO PEOPLE DO WITH MICROCOMPUTER SYSTEMS IN EUROPE?

Monsieur Gilles de la Salle is a young French executive who has formed and developed in a very short time one of the larger "agence immobilier" in Paris. This is the place to go in Paris when looking for an apartment with 3 bedrooms, furnished, located in the "16. arrondissement," priced between 3,000 and 4,000 Francs, and ...And this is where Monsieur de la Salle's problems start.

With his firm's rapid growth he now has an average of 700 locations available for rent. They can be just rooms, or various kinds of apartments and houses, and they are spread over 20 districts in Paris and a large number of suburbs. They can be furnished, unfurnished, with telephone or without (telephones are not readily available in Paris and we know of stories where people waited two years for their installation).

With his growth, he also had to increase his personnel and now employs 10 people in his agency. So the times when somebody remembered just the right place a customer was looking for are gone.

The information concerning each location had to be typed, copied, and distributed to his employees. His staff had to search through a large pile of papers, keeping in mind all the criteria the customer was looking for. If a place was rented, they would still have to inform nine other people about it. Since others were out in the field as well, within a short time, the 10 files were no longer identical. Apartments were rented twice, and some apartments were never offered at all. The success, however, of a renting agency depends on its reputation and fast rental turnover.

This was the moment when Monsieur de la Salle heard about microcomputers. Today he has a microcomputer system installed. The information concerning the available apartments is stored on floppy disks in one centralized file. No more papers, no more copies. His people key in the criteria of an apartment a customer is looking for and within seconds they have the available apartments on the screen. They can also take a print-out if desired. If nothing matches, they can easily offer alternatives, for instance, in an adjacent district, or at a slightly different price. Once an apartment is rented the file is immediately updated. The file also provides a base for statistics, which could not have been done before. Price comparisons, time of turnover, and analyses by districts are just a few examples.

We believe this is an application which could be of interest for any renting agency in larger cities. The program is written in BASIC. For information call us in Paris, 825-8252 or use our Telex: REPTC 270 339 F.

PRICES FOR MICROCOMPUTER PRODUCTS IN EUROPE

One of the factors which influences the buying decision of a potential microcomputer user is the price he has to pay for his equipment in relationship to the benefits he expects in return from the application. While the expectations of the small businessman or the hobbyist are, to a large extent, the same in the U.S. and Europe, the price can be up to twice as high for the user in Europe. What are the reasons?

Let us discuss the factors which determine the end-user price: Dealer price, transportation cost, importation cost, exchange rate considerations, cash flow, computer store cost, sales tax, and the price/volume sensitivity.

With the exception of the dealer price, all of these factors contribute to the significantly higher price in Europe.

A shipment to Europe is generally done air freight, and the cost for the transportation alone runs between

"A splendid performance in three acts"

ACT-I

Known for its dependability, ease of interfacing, utility and affordable price, the ACT-I enjoys its reputation as one of the most popular "glass teletypes" on the market. If your computer system communicates in serial ASCII, the ACT-I could be just the tool you need to get online.

The ACT-I computer terminal manages a 1024 character display organized as 16 lines of 64 characters selected from the standard upper case ASCII set. Receipt of more than 64 characters on a line or the Line Feed code initiates a scroll operation.

STANDARD ACT-I FEATURES INCLUDE:

Switch selectable data rates of: 110, 300, 600, 1200, 2400, 4800, 9600, and 19200 Baud.

Switch selectable UART options: Odd, even, or no parity, one or two stop bits.

Jumper Selectable Interface: RS232C, 20MA current loop or TTL voltage levels.

- Handsome, rugged, lightweight aluminum cabinet
- Standalone operation — absolutely no processor overhead required
- Highly reliable keyboard with two key rollover
- Clear sharp video output signal (RS170 standard) capable of driving any CRT monitor

Price $400. A cursor control / bell option is available for $25.00.

ACT-II

We've added the convenience of an acoustically coupled modem to the economy and performance of the ACT-I to create the ACT-II. Designed to communicate either with remote processors through its modem, or with local computers via its RS232C or 20MA current-loop interfaces, the ACT-II offers versatility unheard of at its low price. The ACT-II (without monitor) slips easily into an attache case (4 × 14 × 11 inches) to commute with you between work and home.

The ACT-II's demodulator employs four stages of active filtering to minimize the bit error rate of the receiver. If you are eager to join the ranks of those who sit at home and enjoy the use of a powerful computer system across town, the ACT-II can be your "password".

As a further convenience feature, the modulator input and demodulator output are available at jacks on the rear of the ACT-II cabinet so that you may link a local serial device (such as a digital casette tape or even your own computer system) to the remote computer through the internal modem.

The ACT-II can be purchased for only $550.00

ACT-IV

If you're looking for a low priced high powered terminal, consider these features which are all standard with MICRO-TERM's ACT-IV:

DISPLAY: Upper and descending lower case characters, 24 lines of 80 characters, and auto-scrolling.

KEYBOARD: Full ASCII with cursor controls and auto-repeat on several keys.

TRANSMISSION MODES: Character by character or "page" mode.

SPECIAL FUNCTIONS: relative and absolute cursor addressing, home up, erase to end of line, erase to end of screen, fixed tabs, report cursor position, and display control characters.

EDITING: in PAGE mode, the user can insert or delete characters on any line and insert or delete lines on the page.

DATA RATE: 300 to 19200 baud (Switch selectable on rear)

The ACT-IVa comes in a compact (briefcase compatible) cabinet without video monitor for $550.

The ACT-IVb comes complete with a 12" monitor and numeric keypad in a single enclosure for $800.

Optional available features: separate printer port (110-9600 baud) $50.

GENERAL INFORMATION:

All MICRO-TERM products are fully assembled, tested and guaranteed for 90 days. The entire MICRO-TERM product line is available from stock at discriminating computer stores or may be purchased directly from the factory. All prices are less monitors (which start at $130.00) F.O.B. St. Louis, Missouri.

VISA and Master Charge Accepted

MTI
MICRO-TERM INC.
P.O. BOX 9387
ST. LOUIS, MO 63117
(314) 645-3656

10 percent and 30 percent of the value of the goods, depending upon volume and weight.

Importation cost, consisting of handling charges and customs duty into the European Economic Community (EEC), add another seven to nine percent to the cost. Some countries outside the EEC have even higher duties, depending upon the ultimate use of the equipment.

The fluctuating exchange rates represent a very special and complex problem. If the importer is from a country which is weak against the dollar, his prices are normally monitored or controlled by a government price control board, not allowing him to follow the currency fluctuation freely. Therefore, he has to set a price which is based on a forecasted average exchange rate, which protects him against further weakening his country's currency.

In case of a "hard-currency" country like Germany, the importer is faced with devaluation of his stock if his country's currency strengthens further. In both cases, the currency risk will contribute to higher prices relative to daily bank exchange rates. With the wide fluctuation of currencies today, this factor adds up to 10 percent to the end-user price.

Sales tax or added value tax varies widely from country to country. In France, the tax is 17.6 percent and in Germany it is 11.5 percent, adding another three to ten percent to the price difference between Europe and the U.S.

Marketing costs of the computer store are also significantly higher, since he is dealing in a different language environment in most cases, and he cannot take advantage of general advertisements of his equipment suppliers.

Although cash flow problems are very similar in the U.S. and Europe, the European importer ties up more money for the same equipment for a longer period of time. The time between the payment and the receipt of goods is considerably longer, and the added value tax, which in itself is already higher than U.S. sales tax, has to be paid at the time of importation. Adding all this up, a 40 to 50 percent higher price can easily be explained. □

WHITE COLLAR MICROCOMPUTER

Vectored from Page 29

increase in cost as labor prices certainly continue to rise.

Many of today's microcomputing product prices are the result of extremely narrow profit margins. Many products have been priced by manufacturers with little business experience, and do not include sufficient allowance for organizational overhead costs, warranty support costs, and adequate distributor and dealer profit margins. Many products are the result of considerable personal sacrifice by personnel highly motivated by the thrill of a new business opportunity in a new field. As these people become more pragmatic, they will demand increased monetary compensation. As microcomputing products become more of real businesses, profit margins and production costs are sure to rise to more realistic levels. In fact, profit margins must increase in order to provide the stable continuing support from financially sound vendors that business computing users need.

True, some prices will drop as technological improvements continue. But few innnovations likely to have major downward price effects are now on the horizon, and new ones take significant time before reaching the market.

Volume increases will also provide a downward influence on prices. However, the local, individualized support necessary as the basis for a good small business microcomputer vendor precludes large volume operations.

DO BUY WHEN READY

The most important reason not to wait to buy your microcomputer is the loss of benefits to your business. Computers can be a real help. The longer you wait, the more benefits you are missing. □

SENSE LINE

By Al Sutton
Chairman, Pomona Valley Computer Club

The April 1977 issue of Popular Electronics magazine contained a feature article on hobby computer clubs. My name was included in the list of Southern California clubs as coordinator for the SCCS Pomona Valley Chapter (I was the only member). As a result of the article, telephone calls poured in from interested hobbyists. The need for a club in this area was clear, so an organizational meeting was held May 19, 1977. The club initially operated as an unofficial Pomona Valley Chapter of the SCCS, but recently changed the name to Pomona Valley Computer Club (PVCC).

At the first meeting, we decided on a very informal club structure, and I volunteered to serve as chairman. A survey of the membership was conducted to determine areas of interest and experience. To my surprise, we found that the 8080 was not the undisputed choice of Pomona Valley hobbyists. The processors used by the initial membership included equal numbers of 8080's, 6800's, and 6502's; a couple of 6100's; a 2650; and numerous minicomputers.

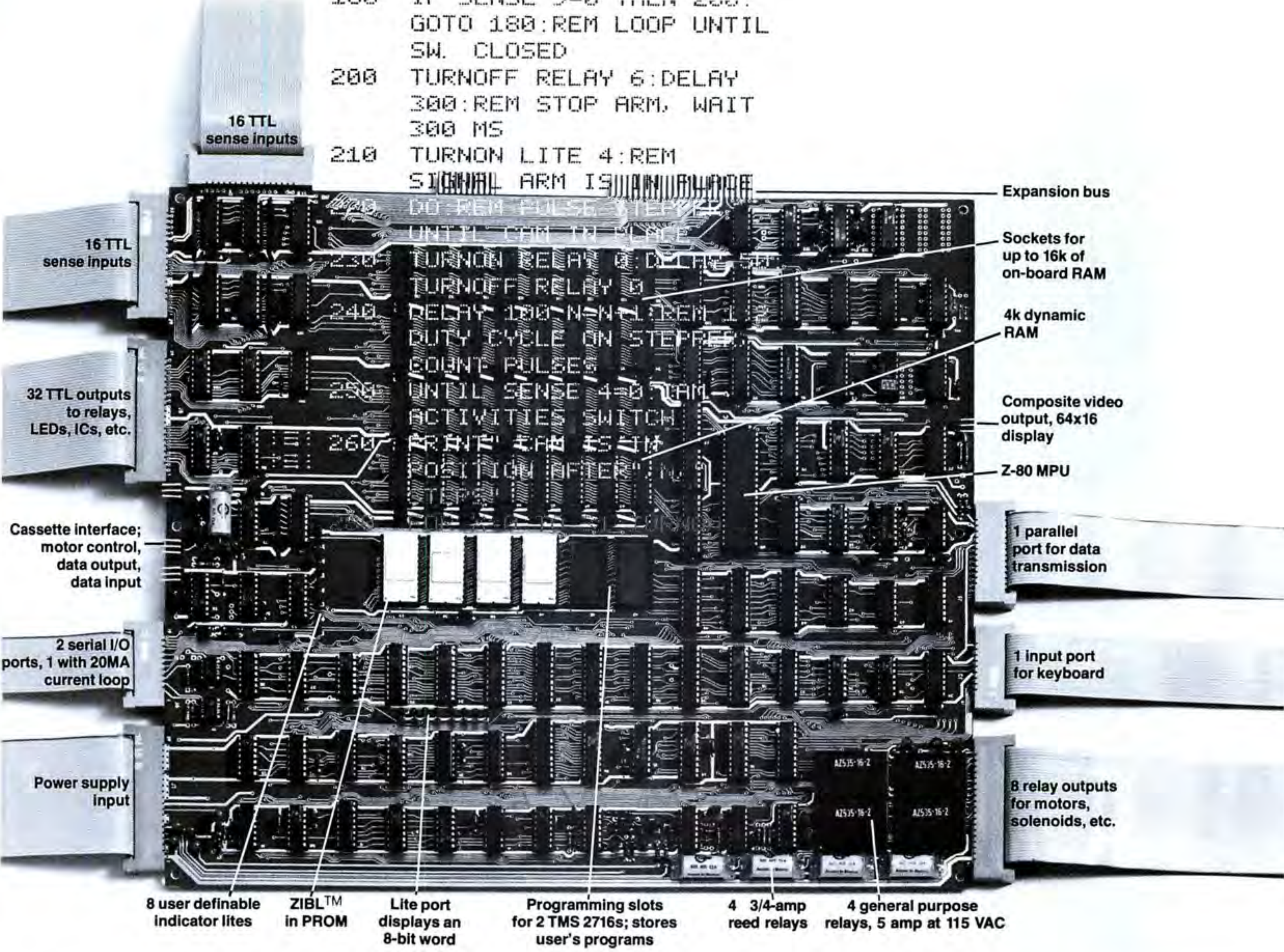
As the club grows, each new member is asked to fill out an information sheet covering occupation, interests, equipment used, and several other categories. These forms are maintained in a central file and are very useful in planning programs and in getting members together when help is needed on hardware or software problems.

The occupations of the members are quite diverse. Although engineers and technicians have a decided edge (19 members), the membership includes a pediatrician, a technical writer, an attorney, a housewife, a social services officer, and several programming, system analysis, and EDP professionals who can't seem to get enough of computers at work. We are also fortunate to have as members several high school and college students and two teachers who are currently offering microcomputer courses at Chaffey College.

Interest categories on the member information sheets are listed below along with the weighted level of interest (0 to 100%) of the membership in each category:

Programming techniques	18.3%
Programming languages	16.7%
Computer architectures	15.0%
Hardware design & construction	16.9%
I/O devices	13.1%
Applications	20.0%

The even spread of interest in all of the categories has made program planning relatively easy, and we have good turnouts for every scheduled program. An important category not covered on the information sheets is "business meetings." This would rate down around four or five percent, since our March 2 meeting (election of officers) had a very small turnout. We did see some of our more dedicated members, and managed to fill some important positions. Our secretary/treasurer, H. Daniel Baernstein, will continue as corresponding secretary/treasurer and will have an assist from our new recording secretary and newsletter editor, Dennis Murphy. We also have a new librarian and hardware coordinator, John Vajgrt. His duties will be to circulate updated listings of books and magazines available to the members, to maintain an equipment advertising bulletin board, and to coordinate our upcoming swap meet (April 16). Our other new officer is our software coordinator,



Dynabyte's new Basic Controller™: Check out its capabilities and imagine your applications

The Basic Controller™ is a powerful, versatile and easy to use single board microcomputer system designed for control applications.

It is heavily into control I/O: relays, flags and sense inputs. What makes controlling these I/Os (and the external devices they control) so easy is our ZIBL™ (Z-80 Industrial Basic Language). It is a superset of NIBL, National Semiconductor's control BASIC, and was written by us specifically for control applications.

We've divided the control world into six categories: sense inputs, flag outputs, lites, relays, A/Ds and D/As. ZIBL implements 64 channels of each in such a way that you need not know anything more about them than their names.

In ZIBL it is valid to say:

100 IF TIME = 053010 AND SENSE (18) = 0 TURNON RELAY 5

Simple, isn't it!

Some but not all of the Basic Controller's mouth watering features include:

- File structures that allow multiple programs written in ZIBL to reside concurrently in RAM. Each program may be individually LOADed, RENAMEd, or RUN. Any program may access another program as though it were a subroutine, while still retaining its own line numbers and variables.
- Complete communication versatility. LISTing, PRINTing and INPUTing may be done to or from any serial or parallel I/O channel or the self-contained CRT I/O.
- Single key SAVE or LOAD to and from cassette.
- Single key SAVE to EPROM. No worry about PROM addressing or programming routines, it is handled by ZIBL — automatically — even if there are other programs already in PROM.
- ZIBL in ROM: TURNON, TURNOFF, DELAY, TIME, REM, IF THEN, DO UNTIL, GOTO, GOSUB, @(exp), TRACE MODE, LINK, READ, DATA, DIR, RND(x,y), strings, triple precision integer arithmetic, plus the usual statements.
- Onboard: Z-80 MPU, 32 flags, 32 sense, 8 relays, 8 lites, 2 serial, 1 parallel, cassette I/O, 64x16 video, keyboard port, two 2716 sockets with programming capability, up to 16k on-board RAM, up to 48k off-board RAM, real time clock, vectored interrupts, Lite Port on board, a kitchen sink, and an Expansion Bus.

$750 assembled, tested, warranted 1 year. You add power supply, keyboard and monitor. Available now — see your computer retailer.

4020 Fabian, Palo Alto, CA 94303; (415) 494-7817

DYNABYTE

CIRCLE INQUIRY NO. 18

Joe Boyle. He will compile a list of all software available for exchange among members, and will try to keep us informed of new software sources. The offices of president, vice-president, and program chairman have not yet been decided, and we will try again at our April 6 meeting, when we hope to have a better turnout. I have been filling these three positions as chairman since the club started, and will welcome the chance to relinquish some of the responsibility.

The people who have provided equipment demonstrations and presentations for our programs deserve special recognition.

Don Ketchum, a member who teaches at Chaffey College, provided an interesting demonstration on his Apple computer and discussed the 6502 architecture and instruction set.

Roger Embree, another member, is currently with Smoke Signal Broadcasting. Roger demonstrated the SSB microfloppy disk system and software, and also gave an excellent presentation on programming with particular emphasis on queuing techniques. Roger is responsible for much of the 6800 software coming from Smoke Signal Broadcasting.

The GNAT computer, a complete 8080 system with dual floppy disks, was demonstrated by Ryall Stewart. The GNAT is an attractive, versatile machine which should serve well in numerous system development, business, text processing, and general purpose computing applications. Ryall is also a member of the club, and a sales representative for GNAT.

Paul Michelson, another member, is currently designing a hush-hush new terminal for Lear Siegler. Paul demonstrated the super-smart LSI VDP-400 video terminal and discussed the basics of smart terminal design. We hope to see Paul's brainchild at a future meeting.

Frank McCoy introduced us to a new tiny language, VTL-2, which is now available for both 6800 and 8080. He also discussed memory test techniques which will be particularly useful to those of us who build and maintain our own hardware.

The intricacies of speech synthesis were discussed and demonstrated by D. Lloyd Write. His COMPUTALKER made a big hit, and his lecture and slides generated considerable interest in speech synthesis techniques.

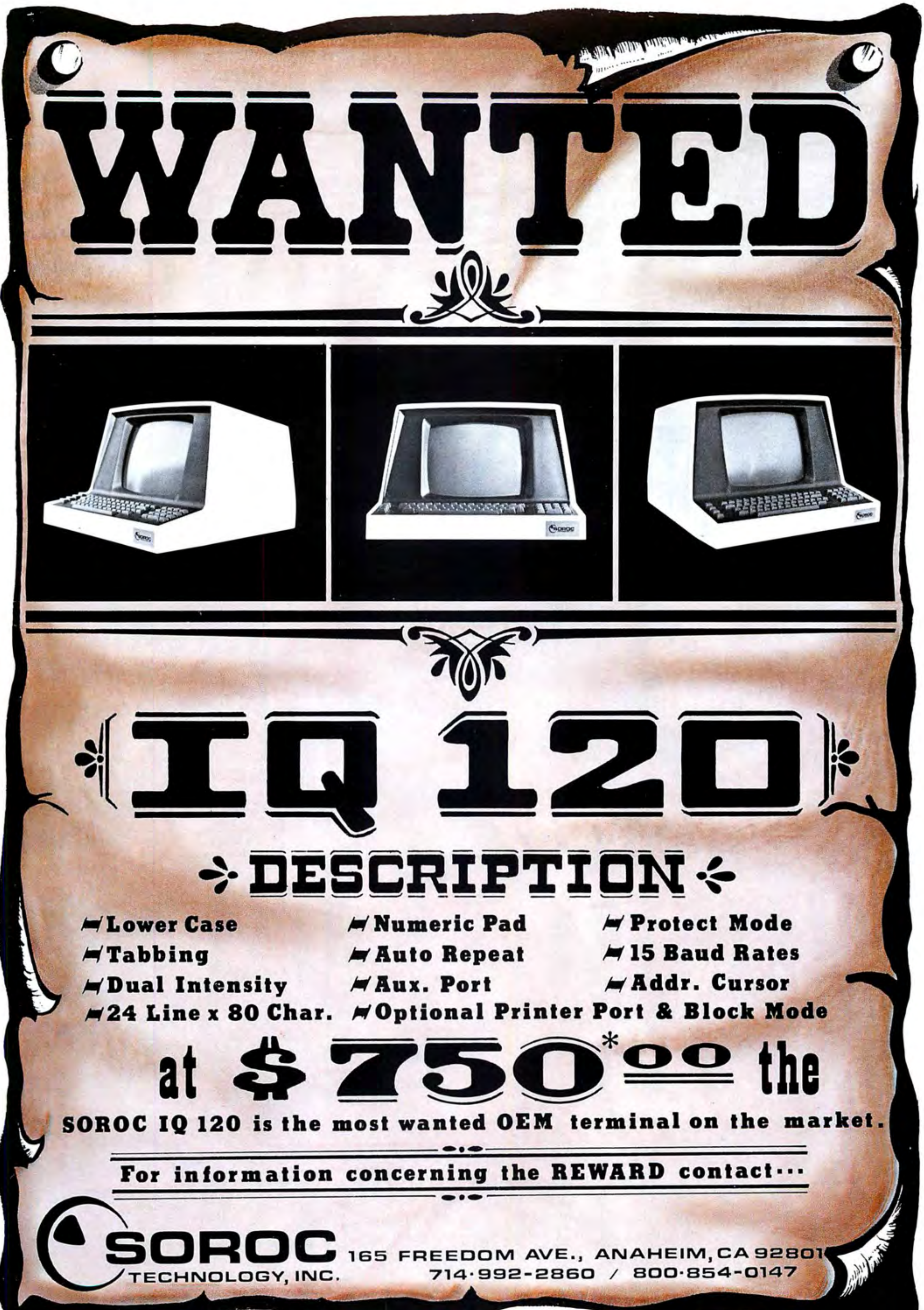
An interesting discussion of the TMS9900 processor architecture and instruction set was provided by Bruce Silveria of Texas Instruments. Bruce also covered the basics of bubble memories, which seem to be hanging heavily over the heads of many manufacturers of other mass storage devices. TI generously provided a TMS9900 chip as a door prize for the meeting.

Future plans for the club include:

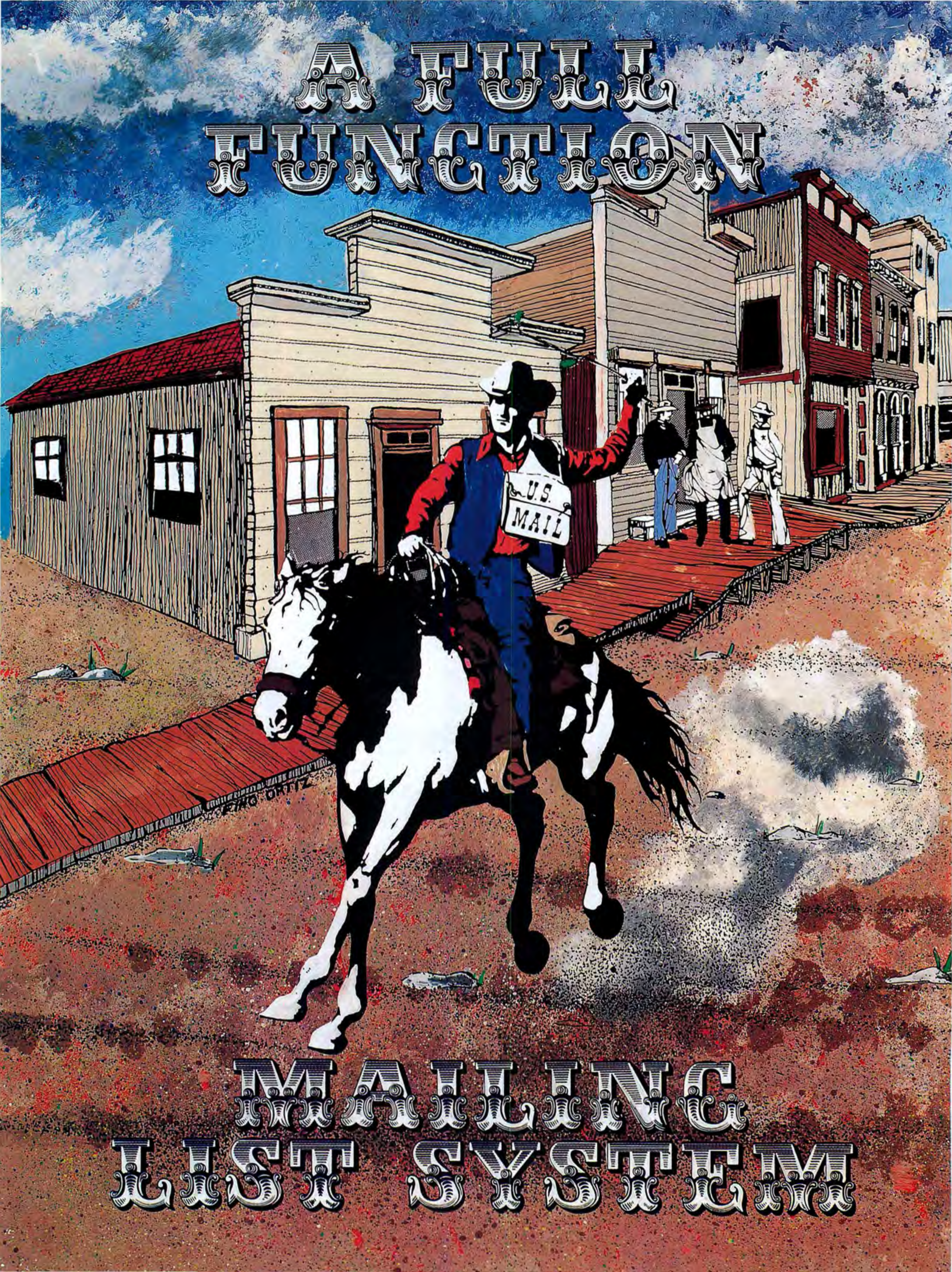
- April 6 — Ed Keith of Citrus College will discuss the fundamentals of structured programming and the use of pseudo-code (7-9 p.m. in the Pomona Public Library).
- April 16 — SWAP MEET in the Perkin-Elmer parking lot, 2771 N. Garey, Pomona (11 a.m. to 3 p.m.).
- May 4 — Tour of Ontario Airport traffic control system (7 p.m.). We will have our meeting at the airport.

Meetings of PVCC are held the first Tuesday of each month from 7 to 9 p.m. at the Pomona Public Library, 625 S. Garey, Pomona, CA. The annual club dues ot $2.50 cover the cost of meeting announcements and brief newsletters mailed prior to each meeting.

If you are interested in joining PVCC, please contact Dan Baernstein at 522 N. Fern Avenue, Ontario, CA 91762. You may reach Dan through his answering service at (714) 983-2723, but make it clear that you are not a patient. If you need further information about PVCC or would like to volunteer your services as a speaker, please contact Al Sutton at 4155 Oak Hollow Road, Claremont, CA 91711, (714) 593-6635. □



* 50-99 QTY./5-49 $800 / 1-4 $995


A FULL FUNCTION
U.S. MAIL
ERIC ORTIZ
MAILING LIST SYSTEM

By John M. Billing

BACKGROUND

Being the only club member with a microcomputer, I was "elected" to be the custodian of the address lists. The "standard" list concept proved to be lacking in some areas. When a member moved, for instance, I had to re-type the whole label into the file. Since the club held competitive events, some of which were open to the public, several lists were required to be maintained, with some names common to more than one list.

Midwest Scientific Instruments released a disk-based version of Robert Uiterwyk's BASIC, called BASIC-2C. This version supports disk data files, and the overlay commands of CHAIN and CALL. CALL is somewhat similar to BASIC as COMMON is to FORTRAN. With this, I undertook to create a system that would fill some of the voids I encountered in the previous list handlers. My intention was to have the system operate on files of up to 1800 labels in as little as 20K of core. Since the interpreter occupies about 14K itself, much use was made of the overlay features to economize the core. This system will run in as little as 19K of contiguous RAM and except-FORMLETR, only uses 18K. The modular concept also permits expansion of the system by merely adding another module with little or no revision to the existing programs.

```
KENNETH          MARKMYWORD, PRES.
GYPSY CONN, INC.
9999 FAR WAY
ANYWHERE          SL 40001


DR. JAMES      W O'BRIEN
MEDICAL ARTS BLDG., SUITE 16
32 HOSPITAL ROAD
CINTUCKY          MO 41213


A. DANIEL        SIDES
133 CORBIN STREET
WINSTON-SALEM    NC 27107


LEE              MALCOLM
5616 FAIRFAX MILL ROAD, APT. 16
ARLINGTON         VA 22333


JOHN             DOE
10 FIRST STREET
PLANETOWN         AA 12345


JANE           A SMITH
1 CENTURY PLACE
NICEBOROUGH       SS 00987


DAVID          W RICHTER
4350 EAST KENNEDY BOULEVARD
VERO BEACH        FL 33609


LAWRENCE       J RIGHTS
826 HIBISCUS
TAMPA             FL 32690


KEN            G JACOBUS
568 LILLIAN DRIVE
DEKALB            FL 60115


JAMES          W JOBORSEN
422 SPRING AVENUE
MADEIRA BEACH     IL 33708
```

SAMPLE RUN

SYSTEM OVERVIEW

Records may be added to or deleted from the data base and records already in the file may be revised in any way. The data base may also be alphabetized, if desired, and can be re-packed eliminating the spaces of the deleted records. A horizontally formatted and paginated record copy may be printed. The labels may be printed on single or quadruple width label stock. Form letters may be produced utilizing the data base for inside addresses. Convenience programs are included for self listing all the modules in one print run and for creating the data files initially.

If your BASIC does not support the CALL function, don't stop reading now, because later on I will give some suggestions for conversion.

The most unique feature of this system is that a list numbering provision is made such that sub-sets of the file may be printed. A single record (label) may be assigned to any one or combination of from one to twenty-four list numbers! This means that counting zero, which is valid, there are 16,777,216 different combinations of list numbers that a label may be assigned. Now the data file may contain totally unrelated lists. A label may be placed in limbo simply by removing any list number(s) it may have, and later, when the activity warrants, the label is again made alive by assigning the appropriate list number(s). Thus, the user of this system is spared the unnecessary re-typing.

Included in the system of program modules is a provision for titling the list numbers. These titles are used by the 'print record copy' module to be printed on the page title line of each page. These titles may be revised at any time and, of course, are stored in a data file.

In addition to the list number categorizing, ZIP number and state flags are provided to create more subsets. If a state flag is set, only those labels whose states match will be printed during that print run. Similary, if a ZIP flag is set, only those labels with matching ZIPs will be printed. Furthermore, the ZIP flag may consist of from one to all five of the ZIP code digits. This flexibility permits lists to be limited to a single post office or to extend over several states. The flags may be used in any combination including none and all. When used in combination, it functions as a logical AND. All flags used must be satisfied in order that the label be printed.

The M.S.I. disk basic handles data files a 256-byte sector at a time, so unless you are willing to accept the wasted disk space, files must be formatted such that the record size is a sub-multiple of 256. For the label file, 128-byte records was chosen. Each record consists of one numeric and eight string variables to facilitate label editing and manipulating. The list code variable is not printed with the label. The middle initial was given its own variable, so that the first name could be used alone in the form letter routine.

The form letter routine uses the same flag system as do the print label modules and further, the return address may be saved for future form letter print runs. If an item changes, such as the date, only that item need be changed. Of course, the body of the form letter is saved and may be edited by line replacement or additions. The file for the body of the form letter has room for 128 lines of 64 characters each. During the operation of this module, the vertical formatting of the letter is automatically handled by the program, adding second and third page titles as necessary. If a label is encountered that has no personal name, the salutation becomes, "Dear Sirs," instead of "Dear (first name)." Space was also allocated for the signer's title, if desired.

The first executable line of each program module is used by the LISTPGMS program to list the module. BYE is the corresponding RETURN to the CALL statement. Note that CALL may not be nexted.

The EXEC program is in reality just a convenience calling the required module as needed. The POKES of lines 30 to 40 permit commas to be entered into string variables. This POKE is not restored because the system exists directly to DOS. If the user forces an exit to BASIC, be aware of this POKE and if modifications to

the system are made using DATA statements, use the open bracket "[" for the data separator.

Before a data file maybe accessed, it must be OPENED, given a PORT number, and its record FIELD must be defined. The OPENFILE Program was included to handle these requirements for each module working with a data file. The variable 0 is set appropriately, and OPENFILE is called to do its thing. The "ON 0 GOTO" of line 20 has all the combinations needed by the system, plus spares for future expansion.

Ten bytes of each record were reserved for the list code. The first version had an ASCII string stored code, but the scheme was rejected because although it could service more list numbers, it was considered too slow in operation. Of course, a call to a USER machine code routine would run at max speed; however, I decided to keep the system in "pure" BASIC, to be compatible with the greatest number of readers. This BASIC handles numeric variables in BCD, packed two to the byte. However, when transferring to or from the disk, it is in ASCII. The encode/decode algorithm used works fine with up to ten digits within BASIC itself, but because of conversion round off errors going to or from the disk, only eight digits could be used reliably.

The alogrithm uses one digit of the code for each three lists in a 1-2-4 code. For example, if record number 489 was to be assigned to lists one and three, the list code would be set to five, and if it were to be on lists one and four, the code would be set to 11. Each digit of the code may have the range from zero to seven. Expanding just the least significant digit of the code for clarity produces the following possibilities:

0 Not on lists 1 or 2 or 3
1 List number 1
2 List number 2
3 Lists 1 and 2
4 List number 3
5 Lists 1 and 3
6 Lists 2 and 3
7 Lists 1 and 2 and 3

Each more significant digit works similarly on a multiple of three. As I mentioned before, there are too many possibilities to list all. The ADDLABEL AND CHNGDATA programs use the encode routine, while the PRNTSNGL, PRNTQUAD, PRNTRCRD, and FORMLETR programs use the decode routine. The CHNGDATA module also uses a decode and print list number scheme. The E(X) array is used as a divisor to select the digit to be decoded and as a multiplier to set the digit to be encoded. In each of these modules, you will notice a routine that sets E(0) to 1, and each successive element to ten times the former. List numbers may be removed by entering its negative. Line 390 of ADDLABEL is an example of the test for a negative entry. The user should be warned that there is no check for a double entry of the same list number. This was not implemented for sake of operating speed. Double entry will give erroneous results! The locations of the encode/decode routines are as follows:

ADDLABEL	**360-470**	
PRNTSNGL	**105-150**	**340-610**
PRNTQUAD	**105-150**	**340-610**
PRNTRCRD	**120-180**	**300-510**
CHNGDATA	**800-930**	
FORMLETR	**105-150**	**300-610**

The printing of the list numbers takes some time, so I caused the terminal to spell out the word, "W O R K I N G", otherwise the operator may think the program has stalled. The control character is sent to return the cursor with no line feed and then six cursor right control characters, (NOTE: not spaces), to position the cursor just before the word, "W O R K I N G". Line 810 accomplishes this for a SWTP 1024 terminal. For other terminals, refer to its instruction manual for the correct control character. Direct cursor control may be substituted if available. "W O R K I N G" is overprinted by the list numbers as they are decoded or if no list number was assigned, the string of spaces of line 825 will wipe it out.

One of my goals was to trap errors before the interpreter saw them, to keep control within the program. Some of these traps are in lines 130-150 of CHNGDATA.

The state flag uses the string variable T$ to store the desired state to be printed. Lines 175, 700, and 710 of PRNTSNGL are an example of the implementation of this flag. If a C/R only was entered signifying no state flag, a zero is stored in T$. This is sensed by line 700, and if not zero, T$ is tested for a match with the state of the label now under consideration.

Table 1. RAM Used By Each Module

EXEC	15972	ADDLABEL	16434	PRNTSNGL	16848
PRNTQUAD	17734	PRNTRCRD	17703	CHNGDATA	18081
ALFABITZ	17765	REVLITI	15432	PRNTITLS	15307
CREATLET	16348	FORMLETR	18928	PACKFILE	16358
OPENFILE	14779	LISTPGMS	15512	FORMAT	16051

As can be seen, the FORMLETR program is the only one over 18K.

The ZIP flag works the same except it measures the length of the ZIP flag entered and compares only these digits. Lines 185, 800, and 810 of PRNTSNGL shows this in action.

The 900 line numbers of PRNTSNGL is the actual print routine which includes a test for blank lines in the label. X sets the total line spacing for each label. It is six in this system. If your label stock requires a different number of lines, an adjustment to the program is necessary. In line 900 of PRNTSNGL, set X equal to the number of required lines minus four. To change PRNTQUAD, add or subtract print statements in lines 960 to 970. Remember, this is a four line label, so this is the minimum!

NOTE: If your BASIC does not support the CALL function, forget about the LISTPGMS module. There is no practical way to implement this without CALL or its equivalent. Each module that calls OPENFILE will have to be revised to include its own open file function. The system will operate fine without CALL at a small core expense. The system could have been written as one single program at great core expense.

The alpha search of CHNGDATA is a simple top-down comparator so that it may be used on unsorted files. It runs slow and was included so that a record may be found when a record file copy had not yet been printed. A record deletion is accomplished by zeroing C and setting all other record variables equal to ASCII spaces. This is done in the 1000 lines. The data of a particular variable is changed in the subroutine at 950. B$ is a holding tank for the record element subject to change and if it is not to be changed, X$ is made equal to B$ and subsequently returned to the file unaltered.

ALFABITZ uses a bubble sort with a short stop tossed in. Again, my motive was the 20K of core. With a large core, more records could be sorted for each disk seek. The storage matrix of lines 120 and 125 could be enlarged and the loader and comparator routines could be changed to suite the larger core. Notice in the sample print out the record without a personal name, floated to the top after the sort. In this sort routine, variables N, I, and J are normal scan limiters. F1 and F2 are set to remember where the last swap was made. The variable F is the

swap-was-made flag, and when a pass completes without a swap, the sort is done. You can save much time here if you enter the records in some semblance of order. Its main purpose is to be able to re-sort the file. If the order is bad and the file is long, it may be days before it finishes the sort.

The T[X] array of PRNTRCRD is used to set the tabs for the printout and the actual tab values are in the data statements, starting at 1100, of only one value per statement. This was done this way for convenience of resetting a tab and not requiring the entire string of numbers to be retyped.

The L[X] array of FORMLETR handles the pagination of the form letters produced. The variable I stored in the file PGMDATA is the line count of the body of the form letter saved by CREATLET and lines 20 to 32 set the L[X] array accordingly. The letters are printed by the lines starting at 900. The subroutine at 1400 prints the body lines, 1500 lines find the page bottom, 1600 lines prints page 2 and/or 3 headings, and 1300 lines prints the close.

The rest of the system's statements are fairly straight forward and should need no further comments. Speaking of which, they were used sparingly in the listing again to save on core.

Table 2. Variables Used in Each Module

EXEC	X,Z
ADDLABEL	C,F,H,O,X,Y,Z,E(11),A$,C$,F$,L$,M$,S$,W$,X$,Z$
PRNTSNGL	C,D,E,F,H,O,X,Z,E(11),A$,C$,F$,L$,M$,P$,S$,T$,V$,W$,X$,Z$
PRNTQUAD	A,B,C,D,E,F,H,X,Y,Z,E(11) A$,C$,F$,L$,M$,P$,S$,T$,V$,W$,X$,Z$,A$(4,4)
PRNTRCRD	C,D,E,F,G,H,L,O,P,Q,R,S,X,Z,E(11),T(11) A$,C$,F$,L$,P$,S$,T$,W$,X$,Z$.
CHNGDATA	C,F,H,M,N,O,X,Y,Z,E(11) A$,B$,C$,F$,L$,M$,U$,V$,W$,X$,Y$,Z$
ALFABITZ	C,F,H,I,J,N,O,X,Z,F1,F2,C(2) C$,F$,L$,M$,S$,W$,X$,Z$ C$(2),F$(2),L$(2),M$(2),S$(2),W$(2),Z$(2)
REVLITI	0,X,Z,I$,X$
PRNTITLS	O,Z
CREATLET	I,O,X,Z,K$,X$
FORMLETR	C,D,E,F,H,I,K,L,O,P,T,T1,T2,X,Z,E(11),L(11) A$,C$,E$,F$,L$,M$,S$,W$,X$,Z$,T$(11)
PACKFILE	C,G,H,O,P,Z,C1 A$,C$,F$,L$,M$,S$,W$,X$,Z$,T$(11)
LISTPGMS	X,Z
FORMAT	D,X,Z,X$
OPENFILE	C,H,I,O,Z A$,C$,E$,F$,G$,I$,K$,L$,M$,N$,O$,Q$,R$,S$,U$,V$,W$,Z$

PROGRAM NUMBER 1

```
0010 REM NAME OPENFILE
0015 IF Z=1 LIST 1-9999:BYE
0020 ON O GOTO 21,22,26,27,31,32,36,37,42,48,52,60,65,70,75,80,85,90,95
0021 OPEN #10, LABELS FOR UPDATE:GOTO 23
0022 OPEN #10, LABELS FOR INPUT
0023 FIELD #10,L$=15,F$=12,M$=1,W$=32,A$=32,C$=15,S$=2,Z$=5,C=10
0024 IF O<9 BYE
0025 RETURN
0026 OPEN #20,TITLIST FOR UPDATE:GOTO 28
0027 OPEN #20,TITLIST FOR INPUT
0028 FIELD #20,I$=51
0029 IF O<9 BYE
0030 RETURN
0031 OPEN #30,PGMDATA FOR UPDATE:GOTO 33
0032 OPEN #30,PGMDATA FOR INPUT
0033 FIELD #30,H=4,I=3,E$=32,G$=32,N$=16,O$=24,Q$=32,R$=32,U$=64,V$=16
0034 IF O<9 BYE
0035 RETURN
0036 OPEN #40,BODYLET FOR UPDATE:GOTO 38
0037 OPEN #40,BODYLET FOR INPUT
0038 FIELD #40,K$=64
0039 IF O<9 BYE
0040 RETURN
0042 GOSUB 21
0044 GOSUB 32
0046 BYE
0048 GOSUB 22
0050 GOTO 44
0052 GOSUB 22
0054 GOSUB 31
0056 GOSUB 37
```

```
0058 BYE
0060 GOSUB 31
0062 GOSUB 36
0064 BYE
0065 GOSUB 27
0067 GOTO 43
7000 REM
7010 REM FOR DUAL DRIVE SYSTEMS, USE
7020 REM 1LABELS IN LINES 21 & 22.
8000 REM
9000 REM OPENFILE
```

```
0010 REM NAME PRNTRCRD (4)
0015 IF Z=1 LIST 1-9999:BYE
0016 Z=24
0020 O=13:CALL OPENFILE
0022 GET #30
0024 FOR X=0 TO 10:READ T(X):NEXT X
0025 PRINT
0026 PRINT "PRINT RECORD COPY, 132 COLUMNS"
0027 PRINT
0028 PRINT
0030 F=B:G=B:L=B:P=B:O=B:R=B:S=B
0035 INPUT "LINE SPACING",X$
0040 IF X$="D" S=1
0050 INPUT "RECORD NUMBERS",X$
0055 IF X$="L" O=1
0060 IF X$="S" O=2
0100 INPUT "LIST # ",J$
0105 IF ASC(J$)<>0 THEN 120
0110 I$="COMPLETE FILE LISTING"
0112 PRINT I$
0115 GOTO 190
0120 E(0)=1
0122 FOR X=1 TO 10
0124 E(X)=E(X-1)*10
0126 NEXT X
0127 IF ASC(J$)<49 THEN 420
0128 IF ASC(J$)>57 THEN 420
0130 X=VAL(J$)
0132 IF X>Z PRINT Z:"M A X !":GOTO 100
0135 SET #20=X
0140 GET #20
0160 E=INT((X-1)/3)
0170 F=X-E*3
0180 E=E+1
0190 INPUT "STATE",T$
0195 IF ASC(T$)<>0 G=1
0200 INPUT "ZIP",P$
0205 IF ASC(P$)<>0 G=1
0210 INPUT "PORT #",X:PORT=X
0230 GET #10
0240 IF LOC#10<H THEN 300
0250 PORT= 1
0260 PRINT
0265 INPUT "ANOTHER LIST",X$
0270 IF LEFT$(X$,1)="Y" THEN 290
0280 CLOSE #10
0281 CLOSE #20
0282 CLOSE #30
0285 CHAIN EXEC
0290 SET #10=1
0295 GOTO 30
0300 IF F=0 THEN 550
0320 C=C/E(E)
0340 D=INT((C-INT(C))*10)
0350 IF D=0 THEN 230
0360 IF D=7 THEN 550
0370 ON F GOTO 400,450,500
0400 IF D/2=INT(D/2) THEN 230
0410 GOTO 550
0420 PRINT "I N V A L I D !":GOTO 100
0450 IF D=1 THEN 230
0460 IF D>5 THEN 550
0470 IF D=2 THEN 550
0480 GOTO 230
0500 IF D>3 THEN 550
0510 GOTO 230
0550 IF ASC(T$)=0 THEN 600
0560 IF T$<>S$ THEN 230
0600 IF ASC(P$)=0 THEN 650
0610 IF P$<>LEFT$(Z$,LEN(P$)) THEN 230
0650 L=L+1
0670 IF L=65 L=Z:PRINT:PRINT:GOTO 650
0680 IF L>4 THEN 900
0690 L=5
0691 PRINT
0692 PRINT
0700 P=P+1
0720 PRINT "PAGE ":P:
0730 PRINT TAB(T(1)):I$:
0750 IF F=0 THEN 770
0760 PRINT TAB(T(5)):"LIST NO. ":J$:
0770 IF G=0 THEN 820
0780 PRINT TAB(T(8)):"FLAG":
0790 PRINT TAB(T(9)):T$:
0800 PRINT TAB(T(10)):P$:
0810 PRINT
0820 PRINT
0900 IF O=0 THEN 930
0910 IF O=1 PRINT LOC#10:
0920 IF O=2 R=R+1:PRINT R
0930 PRINT TAB(T(0)):F$:
0940 PRINT TAB(T(2)):M$:
0950 PRINT TAB(T(3)):L$:
0960 PRINT TAB(T(4)):W$:
0970 PRINT TAB(T(6)):A$:
0980 PRINT TAB(T(7)):C$:
0990 PRINT TAB(T(9)):S$:
1000 PRINT TAB(T(10)):Z$
1010 IF S=1 PRINT:L=L+1
1020 GOTO 230
1100 DATA 6
1110 DATA 15
1120 DATA 19
1130 DATA 21
1140 DATA 41
1150 DATA 70
1160 DATA 74
1170 DATA 107
1180 DATA 110
1190 DATA 123
1200 DATA 126
8000 REM
9000 REM PRNTRCRD (4)
```

```
0010 REM NAME PRNTSNGL  (2)
0020 IF Z=1 LIST 1-9999:BYE
0021 Z=24
0030 O=10:CALL OPENFILE
0040 STRING= 32
0045 E(0)=1
0050 FOR X=1 TO 10
0055 E(X)=E(X-1)*10
```

The New Star In Disk Systems

4 TIMES THE STORAGE. TWICE THE SPEED.

IBM COMPATABILITY

Introducing Thinker Toys™ new DISCUS I™ standard floppy disk system for just $895.*

It's the new standard of performance in S-100 disk systems. Because DISCUS I's standard floppy format holds 4 times the data per disk as North Star's mini-floppy costing $777.

And DISCUS™ is twice as fast as the North Star. And has twice as many tracks. And twice as much data on each track.

That's four times the performance for just $118 more.

But that's not all. DISCUS I™ is both S-100 and IBM-compatible. Plus, our plug-in S-100 controller has an on-board serial I/O port, cache buffer, and read/write/seek routines in on-board ROM. There's even a bootstrap loader.

DISCUS I™ comes as a completely assembled system. A Disk Jockey I™ controller and cable set interface your S-100 mainframe to the handsome free-standing Shugart disk drive unit—which comes complete with cabinet, power supply and fused detachable line cord.

The price also includes DOS, ATE™ assembler/editor, patches for CPM, and Advanced BASIC (available mid-1978) software.

We also offer CPM ($70) and Micro-soft Extended Disc BASIC ($199).

DISCUS I™ has Shugart reliability, Morrow's Micro-Stuff performance and Thinker Toy™ sales and service nationwide. It's fully assembled, tested and warranted (drive 45 days, controller 1 year).

Don't invest in a floppy that can't deliver big performance right now. Buy DISCUS™ standard floppy system today . . . while our special $895 introductory price ($100 savings)* is still available.

Ask your local computer store to order DISCUS I™ before July 1st. Or send your check or money order to THINKER TOYS™, 1201 10th Street, Berkeley, CA 94710. Or call (415) 527-7548, 10-4 pm PST. Add $3.00 handling; California residents add tax.

Thinker Toys™

1201 10th Street
Berkeley, CA 94710

CIRCLE INQUIRY NO. 57

* Introductory prices until June 30, 1978 only.

```
0060 NEXT X
0070 GET #30
0080 V$=CHR$(32)
0100 PRINT "PRINT LABELS, SINGLE WIDTH"
0105 INPUT "LIST# ",X$
0110 IF ASC(X$)=0 F=0:GOTO 170
0111 IF ASC(X$)<49 THEN 420
0112 IF ASC(X$)>57 THEN 420
0115 X=VAL(X$)
0120 IF X>Z PRINT Z:"M A X !":GOTO 105
0130 E=INT((X-1)/3)
0140 F=X-E*3
0150 E=E+1
0170 PRINT
0175 INPUT "STATE",T$
0180 PRINT
0185 INPUT "ZIP",P$
0190 PRINT
0200 INPUT "PORT # ",X
0210 PORT= X
0300 GET #10
0305 IF LOC#10>H THEN 340
0310 PORT= 1
0311 PRINT
0315 INPUT "ANOTHER LIST",X$
0320 IF LEFT$(X$,1)="Y" THEN 335
0325 CLOSE #10:CLOSE#30
0330 CHAIN EXEC
0335 SET #10=1:GOTO 105
0340 IF F=0 THEN 700
0350 C=C/E(E)
0370 D=INT((C-INT(C))*10)
0380 IF D=0 THEN 300
0385 IF D=7 THEN 700
0390 ON F GOTO 400,500,600
0400 IF D/2=INT(D/2) THEN 300
0410 GOTO 700
0420 PRINT "I N V A L I D !":GOTO 105
0500 IF D=1 THEN 300
0510 IF D>5 THEN 700
0520 IF D=2 THEN 700
0530 GOTO 300
0600 IF D>3 THEN 700
0610 GOTO 300
0700 IF ASC(T$)=0 THEN 800
0710 IF T$<>S$ THEN 300
0800 IF ASC(P$)=0 THEN 900
0810 IF P$<>LEFT$(Z$,LEN(P$)) THEN 300
0900 X=2
0910 IF ASC(L$)=32 X=X+1
0920 IF ASC(W$)=32 X=X+1
0930 IF ASC(A$)=32 X=X+1
0940 IF ASC(L$)<>32 PRINT S$:V$:M$:V$:L$
0950 IF ASC(W$)<>32 PRINT W$
0960 IF ASC(A$)<>32 PRINT A$
0970 PRINT C$:V$:S$:V$:Z$
0980 IF X<>0 PRINT:X=X-1:GOTO 980
0990 GOTO 300
8000 REM
9000 REM PRNTSNGL (2)

0010 REM NAME ADDLABEL (1)
0020 IF Z=1 LIST 1-9999:BYE
0021 Z=24
0030 O=9:CALL OPENFILE
0040 E(0)=1
0050 FOR X=1 TO 10
0060 E(X)=E(X-1)*10
0070 NEXT X
0100 GET #30
0110 IF H>LOFF#10 THEN 800
0115 IF H=0 H=1
0120 SET #10=H
0130 CLOSE #30
0160 GET #10
0165 PRINT LOC#10
0170 C=0
0180 INPUT "LAST NAME",L$
0200 IF ASC(L$)=0 THEN 600
0220 INPUT "FIRST NAME",F$
0230 INPUT "INITIAL",M$
0235 INPUT "COMPANY",W$
0240 INPUT "ADDRESS",A$
0260 INPUT "CITY",C$
0280 INPUT "STATE",S$
0300 INPUT "ZIP",Z$
0360 INPUT "LIST # ",X$
0370 F=0
0380 IF ASC(X$)=0 THEN 500
0390 IF ASC(X$)=45 X$=MID$(X$,2):F=1:GOTO 400
0392 IF ASC(X$)<49 THEN 480
0394 IF ASC(X$)>57 THEN 480
0400 X=VAL(X$)
0405 IF X>Z PRINT Z:"M A X !":GOTO 360
0410 Y=INT((X-1)/3)
0420 X=X-Y*3
0430 IF X=3 X=4
0440 X=X*E(Y)
0450 IF F=1 X=-X
0460 C=C+X
0470 GOTO 360
0480 PRINT "I N V A L I D !":GOTO 360
0500 REWRITE #10
0510 GOTO 160
0600 X=LOC#10
0605 CLOSE #10
0610 O=5:CALL OPENFILE
0630 GET #30
0640 H=X
0650 REWRITE #30
0660 CLOSE #30
0670 CHAIN EXEC
0800 PRINT
0810 PRINT "F I L E  I S  F U L L !"
0820 CLOSE #10
0830 GOTO 660
8000 REM
9000 REM ADDLABEL (1)

0010 REM NAME CHNGDATA (5)
0015 IF Z=1 LIST 1-9999:BYE
0016 Z=24
0020 O=9:CALL OPENFILE
0025 GET #30
0030 STRING= 32
0035 E(0)=1
0040 FOR X=1 TO 10
0045 E(X)=E(X-1)*10
0050 NEXT X
0060 U$=""
0100 PRINT
0110 PRINT "CHANGE DATA IN RECORD"
0115 PRINT "C/R ONLY FOR ALPHA SEARCH"
0116 PRINT
0117 PRINT
0120 INPUT "RECORD # ",X$
0130 IF ASC(X$)=0 THEN 200
0131 IF ASC(X$)<48 PRINT"INVALID":GOTO 116
0132 IF ASC(X$)>57 PRINT"INVALID":GOTO 116
0135 X=ABS(INT(VAL(X$)))
0140 IF X=0 THEN 775
0150 IF X>LOFF#10 PRINT"MAX=":LOFF#10:GOTO 116
0160 GOTO 300
0200 PRINT "ALPHA SEARCH:"
0210 INPUT "LAST: ",X$
0220 INPUT "FIRST:",Y$
0225 INPUT "COMPANY",V$
0230 SET #10=1
0240 GET #10
0250 IF LOC#10>=H THEN 940
0260 IF LOC#10=LOFF#10 THEN 940
0270 IF X$<>LEFT$(L$,LEN(X$)) THEN 240
0280 IF Y$<>LEFT$(F$,LEN(Y$)) THEN 240
0285 IF V$<>LEFT$(W$,LEN(V$)) THEN 240
0290 X=LOC#10
0300 SET #10=X
0310 GET #10
0320 IF LOC#10>=H THEN 940
0330 GOSUB 1100
0340 PRINT "LAST NAME: ":L$::B$=L$
0345 INPUT X$:IF ASC(X$)=27 THEN 760
0347 IF X$="D" THEN 1000
0350 GOSUB 960:L$=X$
0355 PRINT "FIRST NAME: ":F$::B$=F$
0360 GOSUB 950:F$=X$
0365 PRINT "INITIAL: ":M$::B$=M$
0370 GOSUB 950:M$=X$
0373 PRINT "COMPANY: ":W$::B$=W$
0375 GOSUB 950:W$=X$
0380 PRINT "STREET: ":A$::B$=A$
0390 GOSUB 950:A$=X$
0400 PRINT "CITY: ":C$::B$=C$
0410 GOSUB 950:C$=X$
0420 PRINT "STATE: ":S$::B$=S$
0430 GOSUB 950:S$=X$
0440 PRINT "ZIP: ":Z$::B$=Z$
0450 GOSUB 950:Z$=X$
0510 GOSUB 800
0520 INPUT X$:F=0
0530 IF ASC(X$)=0 THEN 635
0540 IF ASC(X$)=45 X$=MID$(X$,2):F=1
0542 IF ASC(X$)<49 THEN 615
0544 IF ASC(X$)>57 THEN 615
0550 X=VAL(X$)
0555 IF X>Z PRINT Z:"M A X !":GOTO 520
0560 Y=INT((X-1)/3)
0570 X=X-Y*3
0580 IF X=3 X=4
0590 X=X*E(Y)
0600 IF F=1 X=-X
0610 C=C+X:GOTO 520
0615 PRINT "I N V A L I D !":GOTO 520
0635 INPUT "VERIFICATION",X$
0640 IF LEFT$(X$,1)<>"Y" THEN 755
0720 GOSUB 1100
0740 PRINT
0745 INPUT "CHANGES OK",X$
0750 IF LEFT$(X$,1)<>"Y" THEN 760
0755 REWRITE #10
0760 INPUT "CHANGE NEXT RECORD",X$
0762 IF LEFT$(X$,1)<>"N"THEN 310
0764 INPUT "ANOTHER",X$
0770 IF LEFT$(X$,1)="Y"THEN 120
0775 CLOSE #10
0780 CLOSE #30
0785 CHAIN EXEC
0800 N=0:X=C
0805 PRINT "LIST#  W O R K I N G";
0807 PRINT CHR$(13);
0808 REM FOR ADM-3, USE CHR$(28) IN LINE 810
0810 FOR O=1 TO 6:PRINT CHR$(23)::NEXT O
0815 GOTO 825
0820 N=N+1
0825 IF X=0 PRINT"                          ":RETURN
0830 Y=INT((X/10-INT(X/10))*10)
0840 X=INT(X/10)
0850 M=N*3
0860 IF Y=1 PRINT M+1;
0870 IF Y=2 PRINT M+2;
0880 IF Y=3 PRINT M+1;M+2;
0890 IF Y=4 PRINT M+3;
0900 IF Y=5 PRINT M+1;M+3;
0910 IF Y=6 PRINT M+2;M+3;
0920 IF Y=7 PRINT M+1;M+2;M+3;
0930 GOTO 820
0940 PRINT "N O T  F O U N D !"
0945 GOTO 120
0950 INPUT X$
0960 IF ASC(X$)=0 X$=B$
0970 RETURN
1000 L$=U$:F$=U$:M$=U$:W$=U$
1010 A$=U$:C$=U$:S$=U$:Z$=U$:C=0
1020 PRINT "D E L E T E D !"
1030 GOTO 755
1100 PRINT
1110 PRINT
1120 PRINT "RECORD NUMBER ":LOC#10
1130 PRINT
1140 PRINT F$:TAB(13):M$:TAB(20):L$
1150 PRINT W$
1160 PRINT A$
1170 PRINT C$:TAB(16):S$:TAB(20):Z$
1180 PRINT
1190 GOTO 800
8000 REM
9000 REM CHNGDATA (5)

0010 REM NAME EXEC
0015 IF Z=1 LIST 1-9999:BYE
0020 PORT= 1
0030 POKE( 3513,91)
0035 POKE( 3669,91)
0040 POKE( 4051,91)
0070 LINE= 0
0200 PRINT
0210 PRINT "MAILING LIST SYSTEM.  1.5"
0220 PRINT
0400 INPUT "COMMAND",X
0405 IF X>12 THEN 700
0410 IF X=0 DOS
0490 ON X GOTO 500,510,520,530,540,550,560,570,
     580,590,600,610

0500 CHAIN ADDLABEL
0510 CHAIN PRNTSNGL
0520 CHAIN PRNTQUAD
0530 CHAIN PRNTRCRD
0540 CHAIN CHNGDATA
0550 CHAIN ALFABITZ
0560 CHAIN REVLITI
0570 CHAIN PRNTITLS
0580 CHAIN CREATLET
0590 CHAIN FORMLETR
0600 CHAIN LISTPGMS
0610 CHAIN PACKFILE
0700 PRINT
0710 PRINT X:"IS N O T IN USE!"
0720 PRINT
0730 PRINT "OPTIONS AVAILABLE:"
0740 PRINT
0750 PRINT
0760 PRINT " 0-RETURN TO DOS"
0770 PRINT " 1-ADD LABELS TO DATA BASE"
0780 PRINT " 2-PRINT LABELS, SINGLE WIDTH"
0790 PRINT " 3-PRINT LABELS, QUADRUPLE WIDTH"
0800 PRINT " 4-PRINT RECORD COPY, 132 COL."
0810 PRINT " 5-CHANGE DATA IN RECORDS"
0820 PRINT " 6-ALPHABETIZE DATA BASE"
0830 PRINT " 7-REVISE LIST TITLES"
0840 PRINT " 8-PRINT LIST TITLES".
0850 PRINT " 9-CREATE BODY OF FORM LETTER"
0860 PRINT "10-PRINT FORM LETTERS"
0870 PRINT "11-LIST PROGRAMS"
0880 PRINT "12-PACK DATA BASE"
0980 PRINT
0990 GOTO 400
8000 REM
8100 REM JOHN M. BILLINGS
8200 REM 4101 NELLIE CUSTIS COURT
8300 REM ENGLESIDE, VA 22309
8400 REM
8500 REM 703-780-1688
8600 REM
8700 REM DECEMBER, 1977
8800 REM
9000 REM EXEC

0010 REM NAME PRNTQUAD  (3)
0015 IF Z=1 LIST 1-9999:BYE
0016 Z=24
0020 DIM A$(4,4)
0030 O=10:CALL OPENFILL
0040 STRING= 32
0045 E(0)=1
0050 FOR X=1 TO 10
0055 E(X)=E(X-1)*10
0060 NEXT X
0070 GET #30
0080 V$=CHR$(32)
0100 PRINT "PRINT LABELS, 4 ACROSS"
0105 INPUT "LIST# ",X$
0110 IF ASC(X$)=0 F=0:GOTO 170
0111 IF ASC(X$)<49 THEN 420
0112 IF ASC(X$)>57 THEN 420
0115 X=VAL(X$)
0120 IF X>Z PRINT Z:"M A X !":GOTO 105
0130 E=INT((X-1)/3)
0140 F=X-E*3
0150 E=E+1
0170 PRINT
0175 INPUT "STATE",T$
0180 PRINT
0185 INPUT "ZIP",P$
0190 PRINT
0200 INPUT "PORT # ",X:PORT=X
0210 A=0
0290 FOR X=1 TO 4
0300 GET #10
0310 IF LOC#10=H A=1:GOTO 880
0340 IF F=0 THEN 700
0350 C=C/E(E)
0370 D=INT((C-INT(C))*10)
0380 IF D=0 THEN 300
0385 IF D=7 THEN 700
0390 ON F GOTO 400,500,600
0400 IF D/2=INT(D/2) THEN 300
0410 GOTO 700
0420 PRINT "I N V A L I D !":GOTO 105
0500 IF D=1 THEN 300
0510 IF D>5 THEN 700
0520 IF D=2 THEN 700
0530 GOTO 300
0600 IF D>3 THEN 700
0610 GOTO 300
0700 IF ASC(T$)=0 THEN 800
0710 IF T$<>S$ THEN 300
0800 IF ASC(P$)=0 THEN 820
0810 IF P$<>LEFT$(Z$,LEN(P$)) THEN 300
0820 A$(X,1)=F$+V$+M$+V$+L$
0830 A$(X,2)=W$
0840 A$(X,3)=A$
0850 A$(X,4)=C$+V$+S$+V$+Z$
0860 NEXT X
0880 IF X=1 THEN 1000
0900 FOR X=1 TO 4
0910 FOR Y=1 TO 4
0920 PRINT TAB(33*(Y-1)+1):A$(Y,X);
0930 NEXT Y
0940 PRINT
0950 NEXT X
0960 PRINT
0970 PRINT
0980 IF A=0 THEN 290
1000 PORT= 1
1005 PRINT
1010 INPUT "ANOTHER LIST",X$
1020 IF LEFT$(X$,1)="Y" THEN 1060
1030 CLOSE #10
1040 CLOSE #30
1050 CHAIN EXEC
1060 SET #10=1
1070 GOTO 105
8000 REM
9000 REM PRNTQUAD (3)

0002 REM NAME FORMLETR (10)
0003 PRINT
0004 IF Z=1 LIST 1-9999:BYE
0005 PRINT "PRINT FORM LETTERS"
0006 PRINT
0007 Z=24
0010 O=11:CALL OPENFILE
0011 T=32:T1=10:T2=45
0015 STRING= 64
0018 GET #30
0019 U$=CHR$(44):V$=CHR$(32)
0020 L(2)=0
0022 IF I>=32 L(0)=1:L(1)=0:GOTO 34
0024 IF I>=42 L(0)=I-2:L(1)=2:GOTO 34
0026 L(0)=40
0028 IF I>=82 L(1)=I-40:GOTO 34
```

```
0020 IF I>=92 L(1)=I-42:L(2)=2:GOTO 34
0022 L(1)=50:L(2)=I-90
0034 PRINT
0040 PRINT "RETURN ADDRESS:"
0045 PRINT "     STREET: ":E$:Y$=E$
0047 GOSUB 450:E$=X$
0050 PRINT " CITY,ST.& ZIP: ":G$:Y$=G$
0052 GOSUB 450:G$=X$
0055 PRINT "DATE: ":N$:Y$=N$
0057 GOSUB 450:N$=X$
0060 PRINT "COMPLEMENTARY CLOSE: ":O$:Y$=O$
0062 GOSUB 450:O$=X$
0065 PRINT "SIGNATURE: ":O$:Y$=O$
0067 GOSUB 450:O$=X$
0070 PRINT "TITLE: ":R$:Y$=R$
0072 GOSUB 450:R$=X$
0075 INPUT "SAVE",X$
0080 IF LEFT$(X$,1)="Y" REWRITE#30
0100 PRINT
0105 INPUT "LIST # ",X$
0110 IF ASC(X$)=0 F=0:GOTO 170
0111 IF ASC(X$)<49 THEN 420
0112 E(0)=1:FOR X=1 TO 10:E(X)=E(X-1)*10:NEXT X
0113 IF ASC(X$)>57 THEN 420
0115 X=VAL(X$)
0120 IF X>Z PRINT Z:"M A X !":GOTO 105
0130 E=INT((X-1)/3)
0140 F=X-E*3
0150 E=E+1
0170 PRINT
0175 INPUT "STATE",T$
0180 PRINT
0185 INPUT "ZIP",P$
0190 PRINT
0200 INPUT "PORT # ",X:PORT=X
0300 GET #10
0305 IF LOC#10<H THEN 340
0310 PORT= 1
0311 PRINT
0312 PRINT
0315 INPUT "ANOTHER LIST",X$
0316 PRINT
0317 PRINT
0320 IF LEFT$(X$,1)<>"Y" THEN 330
0325 SET #10=1:GOTO 105
0330 CLOSE #10:CLOSE#30:CLOSE#40
0335 CHAIN EXEC
0340 IF F=0 THEN 700
0350 C=C/E(E)
0370 D=INT((C-INT(C))*10)
0380 IF D=0 THEN 300
0385 IF D=7 THEN 700
0390 ON F GOTO 400,500,600
0400 IF D/2=INT(D/2) THEN 300
0410 GOTO 700
0420 PRINT "I N V A L I D !":GOTO 105
0450 INPUT X$
0460 IF ASC(X$)=0 X$=Y$
0470 RETURN
0500 IF D=1 THEN 300
0510 IF D>5 THEN 700
0520 IF D=2 THEN 700
0530 GOTO 300
0600 IF D>3 THEN 700
0610 GOTO 300
0700 IF ASC(T$)=0 THEN 800
0710 IF T$<>S$ THEN 300
0800 IF ASC(P$)=0 THEN 900
0810 IF P$<>LEFT$(Z$,LEN(P$)) THEN 300
0900 L=21
0910 P=1
0920 K=0
0940 PRINT
0941 PRINT
0942 PRINT
0943 PRINT
0944 PRINT
0945 PRINT
0950 PRINT TAB(T):E$
0960 PRINT TAB(T):G$
0970 PRINT
0980 PRINT TAB(T):N$
0990 PRINT
0991 PRINT
1000 PRINT F$:V$:M$:V$:L$
1005 PRINT W$
1010 PRINT A$
1020 PRINT C$:U$:V$:S$:V$:Z$
1030 PRINT
1035 PRINT
1040 PRINT "DEAR ":
1050 IF ASC(F$)=32 PRINT"SIR":GOTO 1090
1080 PRINT F$
1090 PRINT
1095 PRINT
1100 GOSUB 1400
1110 IF K<=L(0) GOTO 1100
1120 IF L(1)=0 GOTO 1300
1130 GOSUB 1500
1140 GOSUB 1600
1150 GOSUB 1400
1160 IF K<=L(1) GOTO 1150
1170 IF L(2)=0 GOTO 1300
1180 GOSUB 1500
1190 GOSUB 1600
1200 GOSUB 1400
1210 IF K<=L(2) GOTO 1200
1300 PRINT
1305 L=L+7
1310 PRINT TAB(T):O$
1320 PRINT
1321 PRINT
1322 PRINT
1330 PRINT TAB(T):O$
1340 PRINT TAB(T):R$
1350 GOSUB 1500
1360 SET #40=1
1370 GOTO 300
1400 GET #40
1410 PRINT I$
1420 K=K+1
1430 L=L+1
1440 RETURN
1500 FOR X=L TO 65
1510 PRINT
1520 NEXT X
1530 RETURN
1600 PRINT
1601 PRINT
1602 PRINT
1603 PRINT
1604 PRINT
1605 PRINT
```

```
1610 P=P+1
1620 PRINT "PAGE ";P;
1625 IF ASC(L$)=32 PRINT TAB(T1);W$;:GOTO 1640
1630 PRINT TAB(T1);F$;V$;L$;
1640 PRINT TAB(T2);N$
1650 PRINT
1651 PRINT
1652 PRINT
1660 L=10
1670 K=0
1680 RETURN
8000 REM
9000 REM FORMLETR (10)

0010 REM NAME PRNTITLS  (8)
0015 IF Z=1 LIST 1-9999:BYE
0016 Z=24
0020 O=4:CALL OPENFILE
0030 PRINT
0031 PRINT
0040 STRING= 51
0050 PRINT "PRINT LIST TITLES"
0060 PRINT
0120 PRINT
0121 PRINT
0140 INPUT "PORT # ",X:PORT=X
0160 GET #20
0200 PRINT LOC#20;
0220 PRINT TAB(4);
0240 PRINT I$
0250 IF LOC #20 =Z THEN 400
0260 IF X<>1 THEN 160
0280 IF LOC#20<>15 THEN 160
0290 PRINT
0300 INPUT "PRESS C/R TO CONTINUE",I$
0320 GOTO 160
0400 CLOSE #20
0410 CHAIN EXEC
8000 REM
9000 REM PRNTITLS (8)

0010 REM NAME REVLITI  (7)
0015 IF Z=1 LIST 1-9999:BYE
0016 Z=24
0020 O=3:CALL OPENFILE
0040 STRING= 51
0050 PRINT
0051 PRINT
0055 PRINT "REVISE LIST TITLES"
0060 PRINT "ENTER  0  TO CHANGE ALL IN TURN"
0065 PRINT
0120 SET #20=1
0200 INPUT "LIST # ",X
0205 IF X>Z PRINT Z;"M A X ";:GOTO 200
0210 IF X<>0 SET#20=X:GOTO 230
0220 IF LOC #20 =Z THEN 320
0230 GET #20
0240 PRINT LOC#20
0245 PRINT I$
0250 INPUT X$
0260 IF ASC(X$)=0 THEN 290
0270 I$=X$
0280 REWRITE #20
0290 IF X=0 THEN 220
0300 INPUT "ANOTHER TO REVISE",X$
0310 IF LEFT$(X$,1)="Y" THEN 120
0320 CLOSE #20
0330 CHAIN EXEC
8000 REM
9000 REM REVLITI (7)

0010 REM NAME CREATLET  (9)
0020 IF Z=1 LIST 1=9999:BYE
0025 STRING= 64
0030 PRINT
0031 PRINT
0040 PRINT "CREATE BODY OF FORM LETTER"
0041 PRINT
0045 O=12:CALL OPENFILE
0050 INPUT "EDIT MODE",X$
0060 IF LEFT$(X$,1)="Y" THEN 400
0080 GET #30
0100 GET #40
0110 PRINT LOC#40
0120 INPUT K$
0140 IF ASC(K$)<>0 THEN 180
0150 CLOSE #30
0160 CLOSE #40
0170 CHAIN EXEC
0180 REWRITE #40
0190 SET #30=1
0200 GET #30
0210 I=LOC#40+1
0220 REWRITE #30
0230 GOTO 100
0400 PRINT "EDIT BODY OF FORM LETTER"
0420 INPUT "LIST",X$
0430 IF LEFT$(X$,1)<>"Y" THEN 500
0440 INPUT "PORT #",X:PORT=X
0445 PRINT "BODY OF LETTER LISTING"
0446 PRINT
0447 PRINT
0450 SET #40=1
0455 GET #30
0460 GET #40
0470 PRINT LOC#40;K$
0480 IF LOC#40<>I THEN 460
0490 PORT= 1
0500 INPUT "LINE #, TO BE CHANGED",X
0510 IF X=0 THEN 600
0520 SET #40=X
0530 GET #40
0540 PRINT LOC#40;K$
0550 INPUT "NEW LINE",K$
0560 REWRITE #40
0570 GOTO 500
0600 INPUT "ADD MORE LINES",X$
0610 IF LEFT$(X$,1)<>"Y" THEN 150
0620 SET #40=I
0630 GOTO 100
8000 REM
9000 REM CREATLET (9)

0010 REM NAME PACKFILE (12)
0015 IF Z=1 LIST 1-9999:BYE
0020 STRING= 32
0025 PRINT
0028 PRINT
0030 PRINT "REMOVE BLANK RECORDS FORM FILE"
0031 PRINT
0032 PRINT
0035 INPUT "READY",X$
0040 IF LEFT$(X$,1)<>"Y" THEN 900
0100 O=9:CALL OPENFILE
0110 GET #30
0200 GET #10
0210 IF LOC#10=H THEN 840
0220 IF ASC(C$)<>32 THEN 200
0240 P=LOC#10
0260 G=P+1
0270 IF G=H THEN 650
0280 SET #10=G
0300 GET #10:G=G+1
0310 IF ASC(C$)=32 THEN 300
0320 T$(1)=L$:T$(2)=F$:T$(3)=M$:T$(4)=W$
0360 T$(5)=A$:T$(6)=C$:T$(7)=S$:T$(8)=Z$
0400 C1=C
0420 SET #10=P
0440 P=P+1
0460 GET #10
0500 L$=T$(1):F$=T$(2):M$=T$(3):W$=T$(4)
0550 A$=T$(5):C$=T$(6):S$=T$(7):Z$=T$(8)
0600 C=C1
0620 REWRITE #10
0640 IF G<H THEN 280
0650 CLOSE #30
0660 O=5:CALL OPENFILE
0670 GET #30
0680 H=P
0700 REWRITE #30
0710 PRINT
0720 PRINT
0730 PRINT "PACK COMPLETED"
0740 PRINT
0750 PRINT P-1;"RECORDS NOW IN THE FILE"
0760 PRINT
0770 PRINT
0840 CLOSE #10
0850 CLOSE #30
0900 CHAIN EXEC
8000 REM
9000 REM PACKFILE (12)

0010 REM NAME ALFABITZ (6)
0015 IF Z=1 LIST 1-9999:BYE
0020 REM LINES 205,505, & 555 FOR DEMO ONLY
0025 PRINT
0030 PRINT "ALPHABETIZE THE DATA BASE"
0040 INPUT "READY",X$
0050 IF LEFT$(X$,1)<>"Y" THEN 970
0060 O=9:CALL OPENFILE
0100 GET #30
0110 N=H
0120 DIM L$(2),F$(2),M$(2),W$(2),A$(2)
0125 DIM C$(2),S$(2),Z$(2),C(2)
0130 F1=N:F2=N
0200 FOR I=1 TO N-1
0205 PRINT "PASS# ";I,J
0210 F=0
0220 FOR J=1 TO N-I
0230 SET #10=J
0300 FOR X=1 TO 2
0310 GET #10
0320 L$(X)=L$
0330 F$(X)=F$
0335 A$(X)=A$
0340 C$(X)=C$
0345 S$(X)=S$
0350 Z$(X)=Z$
0355 C(X)=C
0360 M$(X)=M$
0365 W$(X)=W$
0380 NEXT X
0400 IF L$(2)>=L$(1) THEN 700
0420 IF L$(2)<>L$(1) THEN 500
0430 IF F$(2)>=F$(1) THEN 700
0440 IF F$(2)<>F$(1) THEN 500
0450 IF M$(2)>=M$(1) THEN 700
0460 IF M$(2)<>M$(1) THEN 500
0470 IF W$(2)>=W$(1) THEN 700
0500 SET #10=J
0505 PRINT :PRINT"SWAP: ";
0510 F=1
0515 F1=J
0520 FOR X=2 TO 1 STEP -1
0530 GET #10
0540 L$=L$(X)
0550 F$=F$(X)
0555 PRINT L$;
0560 A$=A$(X)
0565 M$=M$(X)
0570 C$=C$(X)
0580 S$=S$(X)
0590 Z$=Z$(X)
0600 C=C(X)
0620 W$=W$(X)
0640 REWRITE #10
0650 NEXT X
0660 IF F2=F1+1 THEN 800
0700 IF J>F2 THEN 720
0710 NEXT J
0720 IF F=0 THEN 900
0800 F2=F1
0810 NEXT I
0900 PRINT
0910 PRINT "SORT COMPLETED"
0950 CLOSE #10
0960 CLOSE #30
0970 CHAIN EXEC
8000 REM
9000 REM ALFABITZ (6)

0010 REM NAME LISTPGMS
0015 IF Z=1 LIST 1-9999:BYE
0100 PRINT
0105 PRINT
0110 PRINT "LISTINGS OF ALL PROGRAM MODULES"
0115 PRINT
0120 PRINT
0200 INPUT "PORT #",X
0210 PORT= X
0230 Z=1
0240 PRINT
0241 PRINT
0242 PRINT
0243 PRINT
0244 PRINT
0300 CALL EXEC
0301 PRINT
0302 PRINT
```


INTERFACE AGE ™
MAGAZINE
Presents
THE FLOPPY ROM™
PROGRAM SHEET
33 1/3 RPM
MONAURAL
PLACE COIN HERE IF SOUNDSHEET SLIPS
SIDE 1
A FULL FUNCTION MAILING LIST SYSTEM
By John M. Billing
Written in MSI DOS Version 1.2
Page 64a
INTERFACE AGE Magazine
May 1978, Vol. 3, Issue 5 1978 COPYRIGHT © ALL RIGHTS RESERVED
4478A
MFD. IN U.S.A. BY EVA-TONE® DEERFIELD, ILL.
SOUNDSHEETS


INTERFACE AGE ™
MAGAZINE
Presents
THE FLOPPY ROM™
PROGRAM SHEET
33 1/3 RPM
MONAURAL
PLACE COIN HERE IF SOUNDSHEET SLIPS
SIDE 2
COMPREHENSIVE INDEX TO INTERFACE AGE
December 1975 — December 1976
A USABLE DATABASE
By W.W. Turner & W. Thames
INTERFACE AGE Magazine
Page 64b
May 1978, Vol. 3, Issue 5 1978 COPYRIGHT © ALL RIGHTS RESERVED
4478B
MFD. IN U.S.A. BY EVATONE® SOUNDSHEETS DEERFIELD, ILL.

```
0303 PRINT
0304 PRINT
0305 PRINT
0310 CALL ADDLABEL
0311 PRINT
0312 PRINT
0313 PRINT
0314 PRINT
0315 PRINT
0320 CALL PRNTSNGL
0321 PRINT
0322 PRINT
0323 PRINT
0324 PRINT
0325 PRINT
0330 CALL PRNTQUAD
0331 PRINT
0332 PRINT
0333 PRINT
0334 PRINT
0335 PRINT
0340 CALL PRNTRCRD
0341 PRINT
0342 PRINT
0343 PRINT
0344 PRINT
0345 PRINT
0350 CALL CHNGDATA
0351 PRINT
0352 PRINT
0353 PRINT
0354 PRINT
0355 PRINT
0360 CALL ALFABITZ
0361 PRINT
0362 PRINT
0363 PRINT
0364 PRINT
0365 PRINT
0370 CALL REVLITI
0371 PRINT
0372 PRINT
0373 PRINT
0374 PRINT
0375 PRINT
0380 CALL PRNTITLS
0381 PRINT
0382 PRINT
0383 PRINT
0384 PRINT
0385 PRINT
0390 CALL CREATLET
0391 PRINT
0392 PRINT
0393 PRINT
0394 PRINT
0395 PRINT
0400 CALL FORMLETR
0401 PRINT
0402 PRINT
0403 PRINT
0404 PRINT
0405 PRINT
0410 CALL PACKFILE
0411 PRINT
0412 PRINT
0413 PRINT
0414 PRINT
0415 PRINT
0420 CALL OPENFILE
0421 PRINT
0422 PRINT
0423 PRINT
0424 PRINT
0425 PRINT
0430 CALL LISTPGMS
0431 PRINT
0432 PRINT
0433 PRINT
0434 PRINT
0435 PRINT
0440 CALL FORMAT
0441 REM
0442 REM DELETE LINE 440 IF NOT INCLUDED ON THE DISK
0443 REM
0600 CHAIN EXEC
8000 REM
9000 REM LISTPGMS (11)
```

```
0010 REM NAME FORMAT
0015 IF Z=1 LIST 1-9999:BYE
0020 REM THIS MODULE M U S T BE THE FIRST ONE TO BE EXECUTED!
0030 REM CREATES THE FILES NECESSARY TO THIS SYSTEM
0100 PRINT
0101 PRINT
0102 PRINT
0105 PRINT CHR$(16):CHR$(22)
0110 FOR X=1 TO 10:NEXT X
0120 INPUT "WILL THIS BE USED IN A DUAL DRIVE SYSTEM",X$
0130 IF LEFT$(X$,1)="Y" D=1
0135 PRINT
0136 PRINT
0137 PRINT
0140 PRINT "FORMAT FOR ";
0141 IF D=0 PRINT"SINGLE";
0142 IF D=1 PRINT"DUAL";
0143 PRINT " DRIVE SYSTEMS."
0145 PRINT
0150 PRINT "THE DISK MUST BE PREPARED WITH THE PROGRAMS AS DIRECTED."
0155 PRINT
0156 PRINT
0160 PRINT "IF THIS IS NOT SO, STOP AND DO THIS N O W !"
0165 PRINT
0166 PRINT
0170 INPUT "CONTINUE",X$
0180 IF LEFT$(X$,1)<>"Y" STOP
0200 PRINT "INSERT THE PREPARED DISK INTO DRIVE 0"
0210 INPUT "THEN PRESS C/R",X$
0211 PRINT
0212 PRINT
0215 IF D=1 PRINT"INSERT DATA BASE DISK INTO DRIVE 1"
0216 PRINT
0217 PRINT
0220 PRINT "CREATING THE FILES: PLEASE STAND BY"
0225 PRINT
0226 PRINT
0300 CREATE PGMDATA, REC=255,FILE=1
0310 IF D=1 CREATE 1LABELS, REC=128,FILE=2300:GOTO 340
0320 CREATE LABELS, REC=128,FILE=1800
0340 CREATE TITLIST, REC=51,FILE=30
0360 CREATE BODYLET, REC=64,FILE=128
0500 PRINT "FORMATING COMPLETED."
0505 PRINT
0506 PRINT
0510 PRINT "CONTINUE WITH DISK PREPARATION."
0515 PRINT
0516 PRINT
0517 PRINT
1000 REM THIS MODULE NEED NOT RESIDE ON THE PROGRAM DISK
8000 REM
9000 REM FORMAT
```

MAIL LIST SYSTEM INSTRUCTIONS

Disk Preparation:

Initialize one disk (two if dual drive) using DOS 1.2, marking one 'PROGRAMS 0', and if used, the other 'DATA BASE 1'. To keep seek time to a minimum, save the program modules following this order:

OPENFILE
PRNTRCRD
PRNTSNGL
ADDLABEL
CHNGDATA
EXEC

At this point, LOAD and RUN the module FORMAT following its directions. Then continue saving the rest of the system's modules as follows:

PRNTQUAD
FORMLETR
PRNTITLS
REVLITI
CREATLET
PACKFILE
ALFABITZ
LISTPGMS
BASIC -2C Optinal
FORMAT Optional

This completes disk preparation. The catalog will be shown as:

OPENFILE	55	
PRNTRCRD	55	
PRNTSNGL	55	
ADDLABEL	55	
CHNGDATA	55	
EXEC	55	
PGMDATA	33	
LABELS	33	Single Drive Only
TITLIST	33	
BODYLFT	33	
PRNTQUAD	55	
FORMLETR	55	
PRINTITLS	55	
REVLITI	55	
CREATLET	55	
PACKFILE	55	
ALFABITZ	55	
LISTPGMS	55	
$BASIC(2C)	11	Optional
FORMAT	55	Optional

OPERATING INSTRUCTIONS

With BASIC-2C ready, LOAD and RUN EXEC. Command Number 0 through 12 CALLS appropriate module to accomplish the command, and when finished returns to EXEC to await further commands.

Details of Each Command

1. Add label(s) to the data base.
   The next available record number will be printed on the terminal along with the prompts for the data, thus:

COMPUTER SOFTWARE

for Homeowners, Businessmen, Engineers, Hobbyists, Doctors, Lawyers, Men and Women

We have been in business for over nine years building a reputation for providing a quality product at nominal prices — NOT what the traffic will bear. Our software is:

- **Versatile** — as most programs allow for multiple modes of operation.
- **Tutorial** — as each program is self prompting and leads you through the program (most have very detailed instructions contained right in their source code).
- **Comprehensive** — as an example our PSD program not only computes Power Spectral Densities but also includes FFT's, Inverse-transforms, Windowing, Sliding Windows, simultaneous FFT's variable data sizes, etc. and as a last word our software is:
- **Readable** — as all of our programs are reproduced full size for ease in reading.
- **Virtually Machine Independent** — these programs are written in a subset of Dartmouth Basic but are not oriented for any one particular system. Just in case your Basic might not use one of our functions we have included an appendix in Volume V which gives conversion algorithms for 19 different Basic's; that's right, just look it up and make the substitution for your particular version. If you would like to convert your favorite program in to Fortran or APL or any other language, the appendix in Volume II will define the statements and their parameters as used in our programs.

Over 85% of our programs in the first five volumes will execute in most 8K Basic's with 16K of free user RAM. If you only have 4K Basic, because of its lack of string functions only about 60% of our programs in Volumes I through V would be useable, however they should execute in only 8K of user RAM.

For those that have specific needs, we can tailor any of our programs for you or we can write one to fit your specific needs.

	Vol. II	Vol. III	Vol. IV	Vol. V	
Binomial	Beam	Billing	Bingo	Andy Cap	Quadrac
Chi-Sq.	Conv.	Inventory	Bonds	Baseball	Red Baron
Coeff	Filter	Payroll	Bull	Compare	Regression 2
Confidence 1	Fit	Risk	Enterprise	Confid 10	Road Runner
Confidence 2	Integration 1	Schedule 2	Football	Descrip	Roulette
Correlations	Integration 2	Shipping	Funds 1	Differ	Santa
Curve	Intensity	Stocks	Funds 2	Engine	Stat 10
Differences	Lola	Switch	Go-Moku	Fourier	Stat 11
Dual Plot	Macro		Jack	Horse	Steel
Exp-Distri	Max. Min.		Life	Integers	Top
Least Squares	Navaid		Loans	Logic	Vary
Paired	Optical		Mazes	Playboy	Xmas
Plot	Planet		Poker	Primes	
Plotpts	PSD		Popul	Probal	APPENDIX B
Polynomial Fit	Rand 1		Profits		
Regression	Rand 2		Qubic		
Stat 1	Solve		Rates		
Stat 2	Sphere Trian		Retire		
T-Distribution	Stars		Savings		
Unpaired	Track		SBA		
Variance 1	Triangle		Tic-Tac-Toe		
Variance 2	Variable				
XY	Vector				
APPENDIX A					

Vol. VI	Ledger	Maintains Company accounts and generates financial reports. Includes routines for: Pyrl, Inv, Depr, A/R, A/P.
	ACBS rev: 80	Users Manual
Vol. VII	Chess	Designed to challenge the average player, fairly comprehensive. Great fun for all, offers a unique opportunity for beginners in need of an opponent.
	Medbil	For Doctors and Dentists alike, a complete patient billing system which also permits the maintaining of a patient history record.
	Wdproc	Wordprocessing for lawyers, publishers, writers, etc. Write, store, and change from rough draft to final copy in a variety of formats.
	Utility	Disk utility program with memory testing.
Vol. VIII	1040-Tax	Taxpayers return, itemized deductions or standard
	Balance	Reconciles bank statements
	Checkbook	Balances your checkbook
	Inst1o78	Computes real cost on bank financed items; cars, boats, etc.
	Deprec2	Computes depreciation, 4 methods, any time period

APPENDIX C — FAVORITE PROGRAM CONVERSIONS

Vol. I — $24.95
Bookkeeping
Games
Pictures

Vol. III — $39.95
Advanced Business
Billing, Inventory
Investments
Payroll

Vol. II — $24.95
Math/Engineering
Plotting/Statistics
Basic Statement
Def.

Vol. IV — $9.95
General Purpose

Vol. V — $9.95
Experimenter's
Program

Vol. VI — $49.95
Mini-Ledger

Vol. VII — $39.95
Professional
Programs

Vol. VIII — $19.95
Homeowner's
Programs

SRI

SCIENTIFIC RESEARCH

P.O. Box 490099-I, Key Biscayne, FL 33149
Phone orders call 800-327-6543
Information — (305) 361-1153

Add $1.50 per volume handling, all domestic shipments sent U.P.S. except APO and P.O. Box which go parcel post. Foreign orders add $8.00/volume for air shipment and make payable in U.S. dollars only.

master charge

BANKAMERICARD

AVAILABLE AT MOST COMPUTER STORES

Master Charge and Bank Americard accepted.

Our Software is copyrighted and may not be reproduced or sold.

CIRCLE INQUIRY NO. 47

LAST NAME:	19 Characters	(Extra characters will be ignored)
FIRST NAME:	12	
MID INITIAL:	1	
COMPANY:	32	
STREET:	32	
CITY:	15	
STATE:	2	
ZIP:	5	
LIST #:	0 thru 24	

or any combination of 1 through 24, including all 24. This number is used to print unique lists that will allow a label to appear on more than one list.

In response to any prompt, except 'LAST NAME', a C/R only produces a null string for that item.

A C/R only after 'LAST NAME' signifies completion, and control passes to EXEC.

If a company addressed label without an individual's name is desired, enter a space, then C/R for 'LAST NAME'.

2. Print labels on single width stock. Terminal prompts:

LIST#?	Enter 1 thru 30, or a C/R to deactivate flag.
STATE?	If desired, a two character state abbreviation may be used for a flag.
ZIP?	Any number of ZIP digits may be entered as a print flag. Example: A 3 entered will print every label with a ZIP beginning with a 3, and 22309 entered would allow only 22309 labels to be printed. These flags may be used in any combination and to repeat a C/R only deactivates that flag.
PORT#?	Enter the printers port number. After the list is printed, 'ANOTHER LIST' will appear on the terminal. If 'Y'es is entered, flag prompts will reappear; otherwise EXEC will be called.

3. Print labels on four wide stock.

4. Print record copy, paged and horizontally formatted. Terminal Prompts:

LINE SPACING?	C/R only for single, or 'D' for double spaced lines
RECORD NUMBERS?	C/R only for none, 'L' for file location numbers, and 'S' for sequential numbers.
LIST#$	As described
STATE?	under
ZIP?	command
ANOTHER LIST?	1

5. Change data in existing record. Terminal Prompts:
RECORD NUMBER?
A zero entered will return control to EXEC. A C/R only will invoke an alpha search. Only enough characters need be entered that uniquely identifies the record being sought. All or portions of the LAST, FIRST, and/or the COMPANY names may be used. Here a C/R only omits that name from the search. A record number entered displays the record immediately. Each item of the record will again be shown and wait for a change to be entered. A C/R only will not change this item. Entering new text will replace the item. A SPACE and a C/R erases the item. Entering a 'D' only for the LAST NAME, deletes the record. Pressing the ESC key then a C/R skips this record.

LIST #?
Displays current list numbers for this label. More may be added or, by entering a negative, may be removed. A C/R only response will terminate label revision.

VERIFICATION?
A 'Y'es will display the edited label for approval. Otherwise the label will be stored as revised.

CHANGE NEXT LABEL?
A word starting with N causes the next prompt, or else the next label in the file will be presented for editing.

ANOTHER?
Here a word beginning with Y will prompt for a new record number, or else the program control will be passed to the EXEC module.

6. Alphabetize the data base.
The hierarchy used is last name, first name, initial, and then the company name, in the order of the ASCII code. (A space is first, and numbers are before letters) while this module is running. DO NOT stop it! You may lose the record in the process of being swapped! If your data base is large, now is the time for that picnic.

7. Revise list titles.
Titles of up to 51 characters each may be entered for each list number used, that will be printed on the page prompts are self-explanatory.

8. Print a listing of the list titles. Also self-explanatory.

9. Create the body of the form letter. Terminal Prompts:
EDIT MODE?
Anything entered but a Y will reject the edit mode and assumes a new letter is being created. If a 'Y'es was entered, the new prompt 'LISTING' appears. A 'Y' response causes a query for the 'PORT #?', through which a listing of the existing letter body will be printed.

LINE # TO BE CHANGED?
Enter number(s) as appropriate. A zero will terminate edit mode.

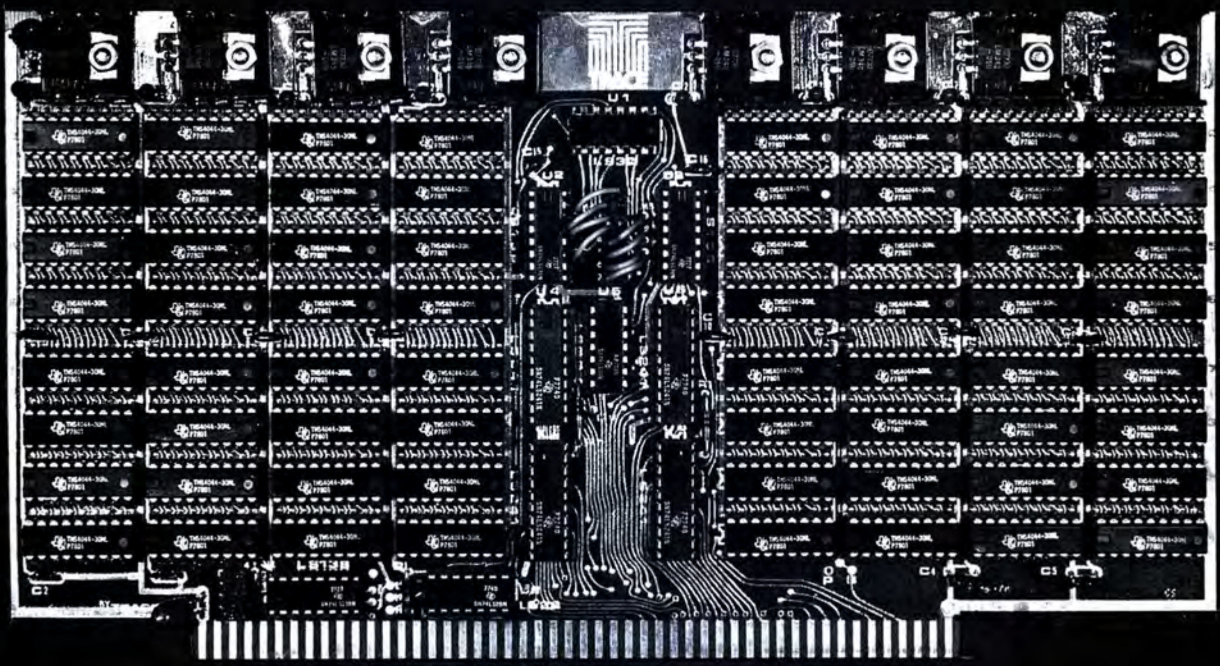
ADD MORE LINES?
If the entry is not a 'Y'es, the letter will be saved and the command module EXEC will be again called. If the entry was a 'Y'es, the next line number is shown. Additional lines may be entered up to a maximum of 123 lines, of 64 characters each. A C/R only here terminates the module.

10. Print form letters.
Prompts are self-explanatory. Note that the return address may be saved and that only those items that are different need be changed. Note also that a new return address may be used for this run, leaving the 'regular' return address undisturbed in the file.

11. Print listing of all program modules.
A convenience module.

12. Repack the file.
Removes blank records and packs the data base without changing the order. □

The author may be contacted by writing to: INTERFACE AGE Magazine, P.O. Box 1234, Cerritos, CA 90701.



Trace Electronics 32K/16K Static RAM

What could you do with this board in your system?

Features

1. Capacity: 32768 Bytes for model 3200
16384 Bytes for model 1600

2. Addressing: Each 4k block separately addressable on any 4k boundary. Allows memory to be placed at the top, bottom, and anywhere in between.
This feature overcomes the problem of working with software that requires memory in different places.

3. Wait states: none

4. Speed: 450 ns. or 250 ns.

5. FULLY STATIC: no clocking, no refreshing.

6. Memory chips: High quality/reliability 4k×1, 18 pin static manufactured by Texas Instruments and second sourced by others.

7. Fully socketed: even the 16k board contains sockets for all 32k or memory.

8. Fully assembled, tested, and burned in.

9. Lower power than equivalent capacity of low power 2102 type memory. And only one slot!

10. DMA Compatible

11. Fully Buffered: All address and data lines buffered with powerful state of the art buffers equipped with Schmitt triggers on their inputs.

12. Special Thermal Design: Each 4k of memory has a separate regulator thereby distributing the heat dissipation over 8 sepatate regulators. They are placed at the top of board to allow the most efficient heat dissipation possible.

13. Megextend TM: up to one megabyte can be addressed providing there is a 4 bit output port in the system. This allows up to 32 model 3200 boards per system, as long as the power requirements are met.
(This feature also makes possible the use of this board in a timeshare environment)

14. Power Required is 1.7A at 8V for model 1600 and 3.3A at 8V for model 3200

Contact your dealer and see this board

Our 32K/16K Static RAM is available right now at many dealers throughout this country. Write Trace Electronics Inc. for a complementary copy of a users manual and the name of the closest dealer carrying our Static RAM.

Trace Electronics Incorporated, P.O. Box 3247, Hampden Station, Reading, Pennsylvania 19604 • (215) 779-3677

TRACE

CIRCLE INQUIRY NO. 58

How To Load Floppy ROM™ #4 and Notes on IAPS™
International ASCII Publication Standard

By Dr. Alan R. Miller, Contributing Editor
W.W. Turner, Southeastern Regional Editor
and Bill Thames

IMPORTANT! READ BEFORE PROCEEDING WITH UNLOADING THE FLOPPY ROM.

Due to the use of 300 baud, only the following programs are recorded on the sound sheet. Side 1, OPENFILE, PRNTRCRD, PRNTSNGL, ADDLABEL, and EXEC. Side 2, DATAFILE.

Attention all 6800, 8080, 6502 and what-ever owners, this Floppy ROM is usable by you! Those of you that have prepared a "Why don't you produce a Floppy ROM for me . . ." form letter with your fancy text editors, may now throw them away. The contents of this Floppy ROM can be unloaded and listed by anyone who has a system capable of running a BASIC language program.

Please note, we said "unload and listed," because, unless the system is running a MSI FD-8 Disk Memory under the control of MSI version 1.2 DOS, the program contained on Floppy ROM Number 4 will not function without some modifications.

For those running a disk oriented computer system, there should be no difficulties in modifying the program. If the system is not disk oriented, it is possible to use the 15 programs, or at least parts of them, to design a mailing list system around a cassette system. The difficulties will be in the area of changing the design of the data files from a random to a sequential organization. This is not an easy task, but it can be done. Regardless of the organization of your system, we recommend that you try to unload the programs.

The data is physically recorded on the Floppy ROM using the familiar Kansas City 300 Baud format. The logical data format was recorded using a new software standard called IAPS. The IAPS format was designed specifically for use in transferring data between two systems, even when the host machines have the following characteristics:

1. Might be based on a different computer architecture such as the Motorola 6800, Intel 8080, Zilog Z80, MOS Technology 6502, or any other popular computer chip.
2. Use different BASIC interpreters or compilers. For instance, in some of the software systems now in use, saved programs are output in a readable ASCII format with control characters imbedded in each line of program code, while other systems save the program in the 'internal' format used by that system. Other systems do not reformat the programs at all, but merely save them as if listed on tape.
3. Most systems provide a method of starting and stopping the tape recorder motors, but some do not provide this under system control. There are systems that use control characters, imbedded in the data, for motor control, while still others require special software to control the cassette system motors.
4. Some systems are very sensitive to control characters in incoming data, even to the point of physically shutting the hardware down.
5. Most systems do not provide any error detection techniques, such as checksums, when they output a saved program to cassette.

Obviously, these differences create some very formidable problems to overcome, when trying to provide quality machine readable programs. Especially when the exact characteristics of the receiving machine are not known!

Here at INTERFACE AGE, it was decided that the best way to handle the problem, was to reformat the "saved" programs into a publishing standard. This would unfortunately require each user to reconvert the distributed programs back into the internal standard used by that user. It was felt, though, that the advantages obtainable by this technique far outweighed the disadvantages of each user having to reconvert the programs upon receipt.

Therefore, we are providing some sample conversion programs for the 6800 and 8080 systems in this article. Access to a 6502 processor was not available, so it was not possible to prepare a conversion program for that system.

This Floppy ROM is a continuation of our attempt to provide quality, usable software to the end user. We will continue to refine the techniques used in the production, distribution, and use of the Floppy ROM's in order to meet that commitment.

The data formatting method used, (IAPS), has been in use for some time now, to transfer data between several systems. Transmission techniques have included the use of homebrew and commercial modems over the telephone network, and mailed cassette tapes.

To be able to recover the data from the Floppy ROM, the ability to read a 300 baud Kansas City "cassette" tape must exist. (The frequencies used are 1200 and 2400 Hz.)

The process is to playback the Floppy ROM, and rerecord the tone signals onto a cassette tape for further processing by the target computer, using the following procedures:

* 1. Rerecord the Floppy ROM onto a cassette tape. Be

*Floppy ROM and IAPS are trademarks of INTERFACE AGE Magazine, Cerritos, CA.

**For those of you who are not quite sure of the techniques required to work with a Floppy ROM, we suggest that you read Orv Balcom's Floppy ROM Loading Techniques in the March and April issues of INTERFACE AGE.*

sure to leave a small pause between each of the 15 individual programs and also before the data base. This is to allow starting and stopping of the recorder program file by program file, while running the conversion program.

2. Load the conversion program for your computer. Make sure you have selected the proper options, with regard to motor control, terminal and cassette recorder addresses.
3. Place the Floppy ROM cassette in the cassette player.
4. Run the program. The computer should display a message "READY TO START" on the terminal.
5. Type a "carriage return." If the tape did not start, or if the system does not have the ability to start the cassette motor, manually start the cassette player.
6. The computer will indicate that it has completely read the tape by responding with "READY TO RECORD."
7. Place a new blank cassette into your cassette recorder. Manually position the tape beyond the tape leader, if any is present.
8. Type a "carriage return." If the output tape did not start, or if the system does not have the ability to start the cassette motor, manually start the cassette recorder.
9. The computer system is now converting the "IAPS" format into a format that the system will accept either as a "saved" program or as a "fast typist" depending on the options selected when creating the desired version of the conversion program.
10. When the computer has finished recording, it will display a "READY TO CONTINUE" message on the terminal.
11. A "carriage return" will cause the system to respond with the "READY TO START" message, so the next program can be converted. Repeat steps 1 to 11 for each of the 15 programs and also for the data base that is recorded on the Floppy ROM, side 2.

When conversion of the program is finished, load BASIC into the computer system. If the 'fast typist' option is selected for conversion, then patch BASIC as necessary to read the cassette tape as keyboard input. Then read the first converted program into the system. Remember, with the 'fast typist' mode of operation, there are no cassette control signals recorded on the cassette tape. It will be necessary to manually stop the tape at the end.

Change any commands that the system will not accept, and resave the program using the BASIC cassette save command (usually "CSAVE" or "TSAVE"). Repeat the above procedure as necessary.

FOR THOSE WITH AN 8080

The 8080 version has five commands:

C	copy IAPS source directly to memory
L	load IAPS source after decoding
D	dump memory to tape
Control-C	correct keyboard error
Control-X	return to monitor

An assembly listing is given in Program 4. The program is started by branching to the beginning (the label START). The program responds by printing the prompt character > on the console.

LOADING A IAPS SOURCE PROGRAM

The IAPS checksummed program can be loaded into memory from cassette or paper tape by typing an L (load), a four-character hexadecimal address where program loading is to begin, and a carriage return. The program is input from the port defined by the names FSTAT/FDATA. After the program is loaded, the console will print out the memory location of the program end. If any checksums occur, the message: (n) CHECKSUM ERRORS will be printed, followed by a list of the corresponding block numbers. If there are no errors, there will be no message. This method of loading will reproduce the original memory image that the IAPS tape was made from. That is, only the data between the STX and ETX is entered. Any character after a DLE character is converted back to its original control character form, and a carriage return is inserted at the end of the record (block), when the ETX character is encountered.

A second method of loading IAPS tapes is also provided. This method is especially useful for making copies of IAPS tapes, since the entire tape is copied exactly into memory. Type a C (copy) followed by the four-character hexadecimal load address and a carriage return. Again at the conclusion of the load, the final address will be printed on the console. As in the first load method, the data are summed during the load process and compared to the checksum bytes at the end of each block. If any of the blocks are improperly read, the above error message will be printed, followed by the actual block numbers in error.

The third command can be used to produce a source tape of the program loaded with the L or C command. Type a D (dump), the starting address, and a carriage return. The file will be output to the tape port (TSTAT/TDATA). The end address is not needed since a binary five was placed at the end of the file when it was originally loaded. After each carriage return is output, a double loop is executed to provide a time delay. If more of a time delay is needed, increase the initialized value of register D at address 8131 HEX.

Two additional commands are available. Typing a Control-C on data entry allows for error recovery. The program restarts, printing the prompt >. When done, return to the monitor, the address defined by the label MONIT at 8161 HEX, by typing a Control-X.

FILE NAME	NO. OF LINES	NO. OF CHARS.	TIME (IN SEC.)
OPENFILE	46	1367	45
PRNTRCRD	135	3343	111
PRNTSNGL	75	2091	70
ADDLABEL	63	1642	54
CHNGDATA	151	4165	138
EXEC	63	1860	62
PRNTQUAD	85	2243	74
FORMLETR	200	4668	155
PRNTITLS	28	740	24
REVLITI	32	888	29
CREATLET	56	1472	49
PACKFILE	54	1436	47
ALFABITZ	75	1940	64
LISTPGMS	127	2395	79
FORMAT	57	1851	61
TOTAL	1247	32101	1070
DATAFILE		7987	266

Table 1. Floppy ROM #4 Timing Chart

GETTING STARTED

The 8080 version requires a little more than one-half K bytes of memory. The stack is placed out of the way, near the top of memory. The console is addressed to 10/11 HEX (20/21 octal) and the tape and file ports are addressed to 12/13 HEX (22/23 octal). Table 2 gives some of the locations and parameters that may have to be changed for a particular system.

The assembly listing shown in Program 4 was pro-

	Source Program Variable	Address (HEX)	Data (HEX)
Define stack	STACK	F3A0	
Console status	CSTAT	814F, 8165	10
Input mask	CIMSK	8151	01
Output mask	COMSK	8167	02
Jump zero		8152, 8168	CA
Tape status	TSTAT	8144	12
Tape data	TDATA	814C	13
Output mask	TMOSK	8146	02
Jump zero		8147	CA
File status	FSTAT	80E4	12
File data	FDATA	80EB	13
Input mask	FIMSK	80E6	01
Jump zero		80E7	CA

Table 2.

duced on a cross assembler that is similar to the TDL macro assembler. All the OP codes are standard, but the pseudo-OPs are a little different. All are preceded by a decimal point. One- and two-byte constants are defined with .BYTE and .WORD respectively. Storage is defined by .BLKB. Global symbols are defined with = =, and the assembly address is defined with a .PHASE directive. Hexadecimal constants are preceded with a $ as with 6800 code.

PRELIMINARY SPECIFICATIONS: International ASCII Publishing Standard (IAPS)

Floppy ROM Number 4 contains 16 separate data transmission files. The first 15 files are part of the MAILING LIST system described elsewhere in this issue. The last transmission file is the INTERFACE AGE cumulative index for 1975-76, Volume 1. Use of this machine readable index will be described beginning in the June issue.

Each data transmission file contains one or more blocks of data in a controlled format. The final block of data is followed by a single /EOT/ character.

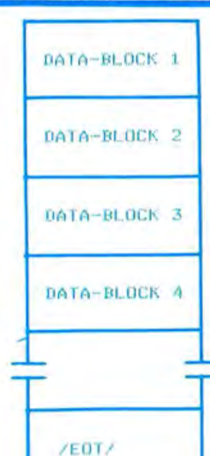


Figure 1. Transmitted Data File

Each of the transmission blocks are variable and can be broken down into three sections. The first section contains a one to eight digit block identification number. The second section is "pure" text and does not contain any control characters other than the /DLE/ character (HEX '10'). The third section has four specially formatted characters that make up a checksum to verify accuracy of the received data.

SECTION IDENTIFICATION

The first section is preceded by a /SOH/ character. A /STX/ character is used as a separator between the first and second sections, while an /ETX/ character is used as a separator between the second and third sections.

There are no separator characters after the four checksum characters, and any characters found there, with the exception of an /EOT/ should be ignored. The next block, if present, will start with a /SOH/ character.

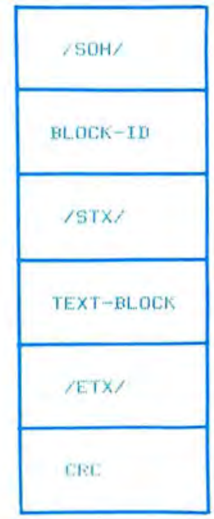


Figure 2. Transmission Block

/SOH/ START-OF-HEADER CHARACTER, (HEX '01')

The START-OF-HEADER character identifies the start of a transmission record. The CRC checksum is reset to zero when a /SOH/ character is detected. The checksum accumulation starts, therefore, with the character that follows the /SOH/ character. Any character that occurs after the checksum characters and prior to a /SOH/ should be ignored unless it is a /EOT/ character.

BLOCK-ID

The BLOCK-ID is a one to eight digit number used to identify each transmission block in the series of blocks that make up a single transmission file. This number is assigned sequentially and starts with 1, as block 0 is reserved for future use.

/STX/ START-OF-TEXT CHARACTER (HEX '02')

The STX character is used to define the start of the text portion of the transmission block.

TEXT-BLOCK

The TEXT-BLOCK contains either a single programming statement or a single data record, depending on which of the 16 files you are looking at. Except for DLE character sequences, all characters in the text portion of each block are printable ASCII characters. There are no other control characters in the text portion of a transmission block except the /DLE/ sequences.

All characters that had a value of less than a space (HEX '20') were translated to a two-byte sequence when the Floppy ROM master was recorded. The first byte of the two-byte sequence will always be a /DLE/ (HEX '10'). The second character of the two-byte sequence is the original character OR'ed with a HEX 40, causing it to be translated into a character sequence of HEX '40' through HEX '5F'.

/ETX/ END-OF-TEXT CHARACTER (HEX '03')

The /ETX/ character marks the end of the TEXT-BLOCK, and is also used as an indicator to identify the start of the checksum characters.

CRC, Checksum

The Checksum characters form a two's complement

16-bit checksum. The checksum includes all characters after the SOH, up to and including the ETX. The checksum characters are then created by separating the 16-bit two's complement sum into four nibbles. Each nibble is then OR'ed with a HEX 40 to create a byte in the HEX 40 to 4F range.

/EOT/ END-OF-TRANSMISSION CHARACTER, (HEX '04')

The /EOT/ character is used to terminate the transmission sequence. It marks the end of the transmission file, even if it appears to occur in the middle of a transmission block. This will allow you to locate the end of a transmission file even when transmission errors occur. Of course, if a transmission error causes the EOT character to not be recognized, there is nothing we can do about it.

COMING ATTRACTIONS

The programs in this article are intended to be the minimum needed to use Floppy ROM Number 4. In the coming months INTERFACE AGE will present programs with additional features. One of these new features will allow for the generation and verification of a tape in the IAPS format with data held in memory. Another feature will prevent loading a tape over this loader program (which would, of course, wipe out the loader and maybe other things as well).

INTERFACE AGE will also be considering other features of error recovery. The course of action on discovery of an error is not always obvious. One possibility is to discontinue further loading and return to the command level when a checksum error is encountered. Another possibility is to print the block number immediately on finding a checksum error. In this case, however, the reading program will fall behind unless output is on a video screen.

The 8080 program given here uses a third approach. It continues without warning until it reaches the end of the file, saving the incorrectly read block numbers in a buffer. Then after the entire file has been read, the block numbers with errors are printed out. Each of these methods has advantages and disadvantages. In the end it may not matter.

Checksum errors may occur with magnetic tape because the tape head is dirty or because the tape is worn out. Any type of checksum error message will alert the user that a problem has arisen.

It is hoped that one or more of the IAPS conversion programs for either the 8080 or the 6800 will assist you in the unloading of the Floppy ROM.

All future Floppy ROMs will be recorded in the IAPS format, so hold onto the conversion programs. When a reliable high speed (over 300 baud) recording method, compatible with the Floppy ROM concept is available, INTERFACE AGE will consider converting from the Kansas City standard to the newer format.

We would appreciate hearing from anyone who either found it necessary to modify or to create another IAPS converison program, or who had difficulty running an existing IAPS conversion program. If you should desire a return answer by mail, please include a self-addressed stamped envelope. Mail to: INTERFACE AGE Magazine, Attn: IAPS, P.O. Box 1234, Cerritos, CA 90701.

Happy converting . . .! □

PROGRAM 1

```
                    NAM     FMIAPS
              *******************
              *
              * WRITTEN BY
              * BILL THAMES, WB4ARN
              *
              * WRITTEN FOR SWTPC
              * 6800 WITH SWTBUG.
              *
              * CONVERTS FROM IAPS
              * TO ANY BASIC WRITTEN
              * BY ROBERT H. UITERWYK
              *
              * REQUIRES AC-30 OR
              * EQUIVALENT.
              *
              *******************
                     OPT    O,S,NOP,NOG
0100                 ORG    $0100
0100 7E 0141         JMP    START
0103 01       SOH    FCB    1
0104 02       STX    FCB    2
0105 03       ETX    FCB    3
0106 04       EOT    FCB    4
0107 05       EOF    FCB    5
0108 0D       CR     FCB    $0D
0109 A042     SPOINT FDB    $A042
010B FFFF     DELAYX FDB    $FFFF
010D 01       DELAYB FCB    1
010E 01FD     BUFFER FDB    END
0110 01FD     BUFNXT FDB    END
0112 0002     INDEX1 RMB    2
0114 1016     MSGIN  FDB    $1016
0116 0000            FDB    0
0118 52              FCC    /READY FOR INPUT/
0127 04              FCB    4
0128 13       MSGOUT FCB    $13,$10,$16
012B 0000            FDB    0
012D 52              FCC    /READY FOR OUTPUT/
013D 04              FCB    4
013E 7E E07E  PDATA1 JMP    $E07E
              *
              * INIT STACK AND
              * BUFFER POINTERS
              *
0141 BE 0109  START  LDS    SPOINT
0144 FE 010E         LDX    BUFFER
0147 FF 0110         STX    BUFNXT
              *
              * OUTPUT 'INPUT' MSG
              *
014A CE 0114         LDX    #MSGIN
014D 8D EF           BSR    PDATA1
              *
              * SET UP PORT #3
              * FOR INPUT/OUTPUT
              *
014F CE 8008         LDX    #$8008
0152 FF A00A         STX    $A00A
              *
              * LOOP UNTIL -CR-
              * TYPED
              *
0155 8D 33    READON BSR    INPUT
0157 81 0D           CMP A  #$0D
0159 26 FA           BNE    READON
              *
              * INIT X-REG AND
              * START READER
              *
015B FE 0110         LDX    BUFNXT
015E 86 11           LDA A  #17
0160 8D 25           BSR    OUTPUT
              *
              * LOOP UNTIL -SOH-
              * OR -EOT- FOUND
              *
0162 8D 26    NXTLIN BSR    INPUT
0164 B1 0106         CMP A  EOT
0167 27 24           BEQ    ENDFIL
0169 B1 0103         CMP A  SOH
016C 26 F4           BNE    NXTLIN
              *
              * BYPASS THE BLOCK
              * ID NUMBER. SKIP
              * EVERYTHING UNTIL
              * -STX- FOUND
              *
016E 8D 1A    NXLIN1 BSR    INPUT
0170 B1 0104         CMP A  STX
0173 26 F9           BNE    NXLIN1
              *
              * SAVE ALL CHARS AFTER
              * -STX- UP TO AND
              * INCLUDING THE -ETX-
              * IN THE BUFFER AREA.
              *
0175 FE 0110         LDX    BUFNXT
0178 8D 10    NXLIN2 BSR    INPUT
017A A7 00           STA A  0,X
017C 08              INX
017D B1 0105         CMP A  ETX
0180 26 F6           BNE    NXLIN2
0182 FF 0110         STX    BUFNXT
              *
              * FOUND -ETX-, NOW GO
              * LOOP UNTIL ANOTHER
              * -SOH- FOUND.
              *
0185 20 DB           BRA    NXTLIN
              *
              * THE FOLLOWING TWO JUMPS
              * JUMP TO 'OUTPUT A CHAR'
              * AND 'INPUT A CHAR'
              * FROM THE A-REG.
              *
0187 7E E1D1  OUTPUT JMP    $E1D1
018A 7E E1AC  INPUT  JMP    $E1AC
              *
              ****************
              *
              * YOU GET HERE
              * WHEN TOTAL FILE
              * HAS BEEN READ INTO
              * THE BUFFER.
              *
              * SAVE THE -EOT- IN
              * THE BUFFER, TURN
              * THE READER OFF, AND
              * OUTPUT 'OUTPUT?' MSG.
```

```
0018D A7 00   ENDFIL STA A  0,X
018F 86 14           LDA A  #20
0191 8D F4           BSR    OUTPUT
0193 CE 0128 OUTMSG LDX     #MSGOUT
0196 8D A6           BSR    PDATA1
              *
              * LOOP FOR A -CR-
              *
0198 8D F0    PNCHON BSR    INPUT
019A 81 0D           CMP A  #$D
019C 26 FA           BNE    PNCHON
              *
              * IS NOW READY TO
              * RECORD TAPE IN
              * SAME FORMAT AS
              * "TSAVE" COMMAND
              * IN BASIC.
              *
              * TURN PUNCH ON...
              *
019E 86 12           LDA A  #18
01A0 8D E5           BSR    OUTPUT
              *
              * MAKE TAPE LEADER...
              *
01A2 8D 39           BSR    DELAY5
              *
              * PAUSE IN BETWEEN
              * BASIC STATEMENTS.
              *
01A4 FE 010E OUTLIN LDX     BUFFER
01A7 8D 3A    OUT2   BSR    DELAY2
              *
              * OUTPUT START OF
              * LINE CODE: -STX-
              *
01A9 B6 0104         LDA A  STX
01AC 8D D9           BSR    OUTPUT
              *
              * OUTPUT BASIC STATEMENT.
              * -EOT- MARKS END OF BUFFER
              * -ETX- MARKS END OF BASIC
              *        STATEMENT. OUTPUT
              *        A CARRAGE RETURN
              *        INSTEAD, FOLLOWED
              *        BY A SLIGHT PAUSE.
              *
01AE A6 00    OUT3   LDA A  0,X
01B0 B1 0106         CMP A  EOT
01B3 27 1C           BEQ    OUT5
01B5 08              INX
01B6 B1 0105         CMP A  ETX
01B9 27 0F           BEQ    OUT4
              *
              * WAS THIS A -DLE-
              * SEQUENCE?
              *
01BB 81 10           CMP A  #$10
01BD 26 07           BNE    OUT32
              *
              * YES, CONVERT NEXT
              * CHARACTER TO
              * ORIGINAL CONTROL
              * CHARACTER.
              *
01BF A6 00           LDA A  0,X
01C1 08              INX
01C2 84 3F           AND A  #$3F
01C4 8A 80           ORA A  #$80
     01C6     OUT32  EQU    *
01C6 8D BF           BSR    OUTPUT
01C8 20 E4           BRA    OUT3
              *
              * A -ETX- WAS FOUND
              * CONVERT TO -CR-
              *
01CA B6 0108 OUT4    LDA A  CR
01CD 8D B8           BSR    OUTPUT
01CF 20 D6           BRA    OUT2
              *
              * THE -EOT- WAS FOUND...
              *
              * OUTPUT UITERWYK'S
              * END OF "TSAVE" -ETX-
              *
              * TURN OFF PUNCH AND
              * RETURN TO SWTBUG.
              *
     01D1     OUT5   EQU    *
01D1 B6 0105         LDA A  ETX
01D4 8D B1           BSR    OUTPUT
01D6 86 14           LDA A  #20
01D8 8D AD           BSR    OUTPUT
01DA 7E E0D0         JMP    $E0D0
              ***********************
01DD 8D 09    DELAY5 BSR    DELAY
01DF 8D 07    DELAY4 BSR    DELAY
01E1 8D 05    DELAY3 BSR    DELAY
01E3 8D 03    DELAY2 BSR    DELAY
01E5 8D 01    DELAY1 BSR    DELAY
01E7 39              RTS
01E8 37       DELAY  PSH B
01E9 FF 0112         STX    INDEX1
01EC F6 010D         LDA B  DELAYB
01EF FE 010B         LDX    DELAYX
01F2 09       DEL1   DEX
01F3 26 FD           BNE    DEL1
01F5 5A              DEC B
01F6 26 FA           BNE    DEL1
01F8 FE 0112         LDX    INDEX1
01FB 33              PUL B
01FC 39              RTS
     01FD     END    EQU    *
                     END
SOH     0103
```

PROGRAM 2

```
                     NAM    S6800
                     OPT    O,S,L,NOP
              *******************
              *
              *      WRITTEN BY
              * BILL TURNER, WB4ALM
              *         AND
              * BILL THAMES, WB4ARN
              *
              * LAST UPDATE 4-1-78
              *
              * FOR THE SWTPC
              * 6800 SYSTEM WITH
              * AC-30 CASSETTE.
              *
              * WILL WORK WITH EITHER
              * MIKBUG OR SWTBUG. IF
              * SWTBUG USED, WILL ALLOW
              * AC-30 TO BE ON DIFFERENT
              * PORT FROM MAIN TERMINAL.
              * TERMINAL AND AC-30 MAY BE
              * ATTACHED TO MP-S OR MP-C
              * BOARDS. MP-L NOT SUPPORTED.
              *
              * THIS PROGRAM WILL
              * CONVERT FROM 'IAPS'
              * TO 'FAST TYPIST'
              * MODE OF OPERATION.
              *
              *    OR
              *
              * WILL DUPLICATE
              * AN 'IAPS' FORMATTED
              * TAPE
              *
              *******************
0100                 ORG    $0100
0100 7E 01C5         JMP    START
0103 0459     BUFFER FDB    END
0105 0000     EBUFF  FDB    0
0107 FFFF     DELAY1 FDB    $FFFF
0109 04       DELAY2 FCB    $4          4-1-78
              *
              * NEXT BYTE (OPSYS)
              * IDENTIFIES OPERATING SYSTEM
              *
010A 00       OPSYS  FCB    0
     0000     SWTBUG EQU    0
     0001     MIKBUG EQU    1
              *
              * PORTYP = 1 FOR MP/C CARD
              * PORTYP = 2 FOR MP/S CARD
              *
010B 01       PORTYP FCB    01          4-1-78
     0001     PORMPC EQU    1           4-1-78
     0002     PORMPS EQU    2           4-1-78
              *
               * TERM AND TAPE ARE
               * USED BY SWTBUG TO
               * IDENTIFY PORT ADDRESS
               * OF AC-30 (TAPE) AND
               * OF THE MAIN TERMINAL.
               * (TERM). DEFAULTS ARE
               * TERM AND TAPE ON SAME
               * PORT --- PORT 1.
              *
010B 8004     TERM   FDB    $8004
010D 8004     TAPE   FDB    $8004
010F 7E E1AC  INEEE  JMP    $E1AC
0112 7E E1D1  OUTEE  JMP    $E1D1
0115 7E E1AC  INCH   JMP    $E1AC
0118 7E E1D1  OUTCH  JMP    $E1D1
011B 7E 03A6  INTAPE JMP    TAPEIN
011E 7E 03CB  OUTAPE JMP    TAPEOU
0121 30       HCKSUM FCC    /0000/
0125 0000     COUNT  FDB    0
0127 00              FCB    0
0128 0000     SAVEX  FDB    0
012A 0000     SETX1  FDB    0
012C 0000     SETX2  FDB    0
              ** FOLLOWING IS POSTIONAL DEPENDENT: ***
012E 20       MSGNG  FCC    /  #/
0131 2D       NUMBER FCC    /--------/
0132 2D
0133 2D
0134 2D
0135 2D
0136 2D
0137 2D
0138 2D
0139 20              FCC    / IS BAD/
013A 49
013B 53
013C 20
013D 42
013E 41
013F 44
0140 04              FCB    $04
              ** END OF POSTIONAL DEPENDENT AREA. **
0141 0000     NUMBX  FDB    0
     0143     MSGST  EQU    *
0143 0D0A            FDB    $0D0A
0145 20              FCC    / READY TO START? /
0146 52
0147 45
0148 41
0149 44
014A 59
014B 20
014C 54
014D 4F
014E 20
014F 53
0150 54
0151 41
0152 52
```

```
0153 54
0154 3F
0155 20
0156 04              FCB    4
          0157  MSGPCH EQU    *
0157 0D0A            FDB    $0D0A
0159 20              FCC    / READY TO RECORD? /
015A 52
015B 45
015C 41
015D 44
015E 59
015F 20
0160 54
0161 4F
0162 20
0163 52
0164 45
0165 43
0166 4F
0167 52
0168 44
0169 3F
016A 20
016B 04              FCB    4
          016C  MSGDUP EQU    *
016C 0D0A            FDB    $0D0A
016E 20              FCC    / TYPE 'D' TO DUPLICATE /
016F 54
0170 59
0171 50
0172 45
0173 20
0174 22
0175 44
0176 22
0177 20
0178 54
0179 4F
017A 20
017B 44
017C 55
017D 50
017E 4C
017F 49
0180 43
0181 41
0182 54
0183 45
0184 20
0185 0D0A            FDB    $0D0A
0187 20              FCC    / OR 'C' TO CONVERT :/
0188 4F
0189 52
018A 20
018B 22
018C 43
018D 22
018E 20
018F 54
0190 4F
0191 20
0192 43
0193 4F
0194 4E
0195 56
0196 45
0197 52
0198 54
0199 20
019A 3A
019B 04              FCB    $04
          019C  MSGF   EQU    *
019C 0D0A            FDB    $0D0A
019E 20              FCC    / ALL DONE --- READY TO CONTINUE? /
019F 41
01A0 4C
01A1 4C
01A2 20
01A3 44
01A4 4F
01A5 4E
01A6 45
01A7 20
01A8 2D
01A9 2D
01AA 2D
01AB 20
01AC 52
01AD 45
01AE 41
01AF 44
01B0 59
01B1 20
01B2 54
01B3 4F
01B4 20
01B5 43
01B6 4F
01B7 4E
01B8 54
01B9 49
01BA 4E
01BB 55
01BC 45
01BD 3F
01BE 20
01BF 04              FCB    04
                *
                * INIT STACK POINTER
                * (REPLACE NOP'S
                *   IF NECESSARY)
                * ISSUE 'INPUT?' MSG,
                * WAIT FOR -CR-
                *
01C0 01         START  NOP
01C1 01                NOP
01C2 01                NOP
01C3 FE 0103           LDX    BUFFER
01C6 FF 0105           STX    EBUFF
01C9 CE 0143           LDX    #MSGST
01CC BD 0384           JSR    PDATA1
01CF BD 037C           JSR    WAIT
                *
                * TURN TAPE ON
                *
01D7 FE 0103           LDX    BUFFER
01DA 86 01      TONGET LDA A  #1        4-1-78
01DC BD 0411           JSR    TAPEON
                *
                * GET CHAR, LOOP
                * UNTIL -SOH-
                * FOUND.IF -EOT-
                * ASSUME END-OF-FILE.
                *
01DA BD 011B    GET1   JSR    INTAPE
01DD 81 01             CMP A  #$01
01DF 26 F9             BNE    GET1
01E1 20 0B             BRA    GOTSOH
01E3 BD 011B    GETSOH JSR    INTAPE
01E6 81 04             CMP A  #$04
01E8 27 2B             BEQ    GOTEOT
01EA 81 01             CMP A  #$01
01EC 26 F5             BNE    GETSOH
                *
                * WE GOT A -SOH-
                * RESET CHECKSUM COUNT
                * CLEAR BLOCK-ID FIELD.
                *
01EE A7 00      GOTSOH STA A  0,X
01F0 08                INX
01F1 7F 0125           CLR    COUNT
01F4 7F 0126           CLR    COUNT+1
01F7 FF 0128           STX    SAVEX
01FA CE 0131           LDX    #NUMBER
01FD FF 0141           STX    NUMBX
0200 CE 2020           LDX    #$2020
0203 FF 0131           STX    NUMBER
0206 FF 0133           STX    NUMBER+2
0209 FF 0135           STX    NUMBER+4
020C FF 0137           STX    NUMBER+6
020F FE 0128           LDX    SAVEX
0212 BD 011B    GET    JSR    INTAPE
                *
                * SAVE ALL CHARS IN
                * BUFFER.
                *
0215 A7 00      GOTEOT STA A  0,X
0217 08                INX
                *
                * ADD CHAR TO CHECKSUM
                *
0218 36                PSH A
0219 BB 0126           ADD A  COUNT+1
021C B7 0126           STA A  COUNT+1
021F 24 03             BCC    CKEOT
0221 7C 0125           INC    COUNT
0224 32         CKEOT  PUL A
                *
                * WAS CHAR AN -EOT-
                *
0225 81 04             CMP A  #$04
0227 26 03             BNE    CKSTX
0229 7E 02CE           JMP    ENDIN
                *
                * WAS CHAR AN -STX-?
                *
022C 81 02      CKSTX  CMP A  #$02
022E 26 0C             BNE    CKETX
0230 FF 0128           STX    SAVEX
                *
                * -STX- FOUND, DON'T
                * SAVE ANY MORE
                * BLOCK-ID CHARS
                *
0233 CE 0000           LDX    #0
0236 FF 0141           STX    NUMBX
0239 7E 02AB           JMP    SNUMB1
                *
                * -ETX-? IF SO
                * CALCULATE CHECKSUM
                * AND COMPARE TP
                * TRANSMITTED VALUE.
                *
023C 81 03      CKETX  CMP A  #$03
023E 26 68             BNE    SNUMB
0240 FF 012A           STX    SETX1
                *
                * SAVE XMITTED VALUE.
                *
0243 BD 011B           JSR    INTAPE
0246 A7 00             STA A  0,X
0248 08                INX
0249 BD 011B           JSR    INTAPE
024C A7 00             STA A  0,X
024E 08                INX
024F FF 012C           STX    SETX2
0252 BD 011B           JSR    INTAPE
0255 A7 00             STA A  0,X
0257 08                INX
0258 BD 011B           JSR    INTAPE
025B A7 00             STA A  0,X
025D 08                INX
                *
                * PRODUCE 2'S
                * COMPLEMENT, AND
                * AND CONVERT TO
                * 'ASCII' REPRESENTATION.
                *
025E F6 0126           LDA B  COUNT+1
0261 B6 0125           LDA A  COUNT
0264 50                NEG B
0265 24 01             BCC    CKETX1
0267 4C                INC A
```

```
026D 40         CKETX1 NEG A
026E F7 0128           STA B   COUNT+2
0271 FF 0129           STX     SAVEX
0274 CE 0122           LDX     #HCKSUM
0277 BD 037D           JSR     OPACK
027A B6 0128           LDA A   COUNT+2
027D CE 0124           LDX     #HCKSUM+2
0280 BD 037D           JSR     OPACK
                *
                * COMPARE IT NOW
                *
0283 CE 0122           LDX     #HCKSUM
0286 A6 00             LDA A   0,X
0288 E6 01             LDA B   1,X
028A FE 012B           LDX     SETX1
028D A1 00             CMP A   0,X
028F 26 33             BNE     NOGOOD
0291 E1 01             CMP B   1,X
0293 26 2F             BNE     NOGOOD
0295 CE 0124           LDX     #HCKSUM+2
0298 A6 00             LDA A   0,X
029A E6 01             LDA B   1,X
029C FE 012D           LDX     SETX2
029F A1 00             CMP A   0,X
02A1 26 21             BNE     NOGOOD
02A3 E1 01             CMP B   1,X
02A5 26 1D             BNE     NOGOOD
02A7 FE 0129           LDX     SAVEX
                *
                * THIS BLOCK WAS OK
                * GO GET ANOTHER...
                *
02AA 7E 01E8           JMP     GETSOH
02AD FF 0129    SNUMB  STX     SAVEX
02B0 FE 0146    SNUMB1 LDX     NUMBX
                *
                * ARE WE SAVING THIS CHAR
                * AS PART OF THE
                * BLOCK-ID?
                *
02B3 8C 0000           CPX     #0
02B6 27 06             BEQ     ENUMB
02B8 A7 00             STA A   0,X
02BA 08                INX
02BB FF 0146           STX     NUMBX
02BE FE 0129    ENUMB  LDX     SAVEX
02C1 7E 0217           JMP     GET
                *
                * BLOCK WAS NO GOOD
                * ISSUE MESSAGE.
                *
02C4 FF 0105    NOGOOD STX     EBUFF
02C7 86 04             LDA A   #$04
02C9 A7 00             STA A   0,X
02CB F6 010B           LDA B   PORTYP
02CE D1 01             CMP B   PORMPC
02D0 26 07             BNE     NG1
02D2 86 FF             LDA A   #$FF
02D4 B7 A00C           STA A   $A00C
02D7 20 05             BRA     NG2
02D9 86 3C      NG1    LDA A   #$3C
02DB B7 8007           STA A   $8007
02DE 86 01      NG2    LDA A   #1
02E0 BD 0426           JSR     TAPEOF
02E3 CE 012F           LDX     #MSGNG
02E6 BD 0398           JSR     PDATA1
02E9 20 03             BRA     ENDIN1
                *******************
                * OUTPUT BUFFER:   *
                *******************
     02EB       ENDIN  EQU     *
02EB FF 0105           STX     EBUFF
                *
                * TURN READER OFF
                * AND WAIT FOR OK
                * TO PROCEED
                *
02EE 86 01      ENDIN1 LDA A   #1
02F0 BD 0426           JSR     TAPEOF
02F3 CE 015C           LDX     #MSGPCH
02F6 BD 0398           JSR     PDATA1
02F9 BD 0390           JSR     WAIT
                *
                * OPTION: 'D' OR 'C'?
                *
02FC CE 0171    OPTION LDX     #MSGDUP
02FF BD 0398           JSR     PDATA1
0302 BD 0110           JSR     INEEE
0305 81 44             CMP A   #'D
0307 27 49             BEQ     PUT5
0309 81 43             CMP A   #'C
030B 27 02             BEQ     PUT0
030D 20 ED             BRA     OPTION
                *
                * CONVERSION OPTION
                * SELECTED...
                * TURN PUNCH ON
                * CREATE LEADER
                *
     030F       PUT0   EQU     *
030F 86 02             LDA A   #2
0311 BD 0411@          JSR     TAPEON
0314 BD 03A5           JSR     DELAY3
0317 FE 0103           LDX     BUFFER
                *
                * OUTPUT EACH DATA LINE
                * WITH A TIME DELAY
                * AFTER THE -CR-
                * STOP WHEN -EOT-
                * FOUND
                *
                **********
                * SKIP ALL DATA UP TO
                * AND INCLUDING THE
                * -STX-
                *
031A A6 00      PUT    LDA A   0,X
031C 08                INX
031D 81 04             CMP A   #$04
031F 27 48             BEQ     DONE
0321 81 02             CMP A   #$02
0323 26 F5             BNE     PUT
                *
                * OUTPUT ALL CHARS
                * UNTIL -ETX-
                *
                * DON'T FORGET THE
                * -DLE- SEQUENCES.
                *
                * EVERYTHING BETWEEN
                * -STX- AND -ETX- IS
                * PART OF THE BASIC
                * STATEMENT.
                *
0325 A6 00      PUT1   LDA A   0,X
0327 08                INX
                *
                * -EOT- CHAR?
                *
0328 81 04             CMP A   #$04
032A 27 1C             BEQ     PUT3
                *
                * -ETX- CHAR?
                *
032C 81 03             CMP A   #$03
032E 27 0E             BEQ     PUT2
                *
                * -DLE- CHAR?
                *
0330 81 10             CMP A   #$10
0332 26 05             BNE     PUT11
                *
                * WAS -DLE- ...
                * CONVERT NEXT CHAR
                * BACK TO ORIG.VALUE
                *
0334 A6 00             LDA A   0,X
0336 08                INX
0337 84 3F             AND A   #$3F
                *
                * OUTPUT CHAR
                * IN A-REG
                *
0339 BD 011F    PUT11  JSR     OUTAPE
033C 20 E7             BRA     PUT1
                *
                * -ETX- WAS FOUND
                * OUTPUT A -CR-
                * INSTEAD. THEN
                * PAUSE (ALLOWS
                * RECEIVING SYSTEM
                * TO ECHO A -LF-
                * WITHOUT INTERFERING
                * WITH INCOMING DATA.
                *
033E 86 0D      PUT2   LDA A   #$0D
0340 BD 011F           JSR     OUTAPE
0343 BD 03AE           JSR     DELAYS
0346 20 D2             BRA     PUT
                *
                * -EOT- WAS FOUND
                * OUTPUT FINAL -CR-
                *
0348 86 0D      PUT3   LDA A   #$0D
034A BD 011F           JSR     OUTAPE
034D BD 03AE           JSR     DELAYS
0350 20 17             BRA     DONE
                *
                * DUPLICATION OPTION
                * SELECTED...
                *
                * TURN PUNCH ON
                * CREATE LEADER
                *
0352 86 02      PUT5   LDA A   #2
0354 BD 0411           JSR     TAPEON
0357 BD 03A5           JSR     DELAY3
035A FE 0103           LDX     BUFFER
                *
                * OUTPUT ENTIRE BUFFER
                * TO TAPE --- STOP WHEN
                * -EOT- CHAR FOUND.
                *
035D A6 00      PUT6   LDA A   0,X
035F 08                INX
0360 BD 011F           JSR     OUTAPE
0363 81 04             CMP A   #$04
0365 27 02             BEQ     DONE
0367 20 F4             BRA     PUT6
                *
                * TURN PUNCH OFF
                * WAIT FOR OK
                * TO PROCEED
                *
0369 BD 03A5    DONE   JSR     DELAY3
036C 86 02             LDA A   #2
036E BD 0426           JSR     TAPEOF
0371 CE 01A1           LDX     #MSGF
0374 BD 0398           JSR     PDATA1
0377 BD 0390           JSR     WAIT
037A 7E 01C5           JMP     START
                ********************
                * SUBROUTINES:     *
                ********************
                *
                * CONVERT BYTE IN
                * A-REG TO A 2-BYTE
                * ASCII SEQUENCE OF
                * HEX '40' THRU '4F'
                *
037D 36         OPACK  PSH A
037E 44                LSR A
037F 44                LSR A
0380 44                LSR A
0381 44                LSR A
0382 84 0F             AND A   #$0F
0384 8A 40             ORA A   #$40
0386 A7 00             STA A   0,X
0388 32                PUL A
0375 84 0F             AND A   #$0F
0377 8A 40             ORA A   #$40
0379 A7 01             STA A   1,X
037B 39                RTS
                ********************
                * WAIT FOR -CR- REPLY
                *
     037C       WAIT   EQU     *
037C BD 010F           JSR     INEEE
037F 81 0D             CMP A   #$0D
0381 26 F9             BNE     WAIT
0383 39                RTS
                ********************
                * OUTPUT ALL CHARS
                * OF A MESSAGE UP
                * TO BUT NOT INCLUDING
                * A HEX '04' CHAR.
                * X-REG POINTS TO
                * START OF MESSAGE
                * UPON ENTRY.
                *
0384 A6 00      PDATA1 LDA A   0,X
0386 81 04             CMP A   #$04
0388 27 06             BEQ     PDATA2
038A BD 0112           JSR     OUTEE
038D 08                INX
038E 20 F4             BRA     PDATA1
0390 39         PDATA2 RTS
                ********************
                * SOFTWARE TIMING
                * DELAY LOOP.
                * CHANGE VALUES OF
                * DELAY1 AND DELAY2
                * TO CHANGE LENGTH
                * OF DELAY
                *
0391 FF 0128    DELAYS STX     SAVEX
0394 FE 0107           LDX     DELAY1
0397 F6 0109           LDA B   DELAY2
 39A 5A         DEL1   DEC B
039B 27 05             BEQ     DEL3
039D 09         DEL2   DEX
039E 26 FD             BNE     DEL2
03A0 20 F8             BRA     DEL1
03A2 FE 0128    DEL3   LDX     SAVEX
03A5 39                RTS
                ********************
                * GET A CHAR FROM
                * AC-30 CASSETTE
                * INTO THE A-REG.
                *
03A6 FF 03C9    TAPEIN STX     TX
03A9 F6 010A           LDA B   OPSYS
03AC D1 01             CMP B   MIKBUG
03AE 27 06             BEQ     TAPEI2
03B0 FE 010D           LDX     TAPE
03B3 FF A00A           STX     $A00A
03B6 BD 0115    TAPEI2 JSR     INCH
03B9 F6 010A           LDA B   OPSYS
03BC D1 01             CMP B   MIKBUG
03BE 27 06             BEQ     TAPEI3
03C0 FE 010B           LDX     TERM
03C3 FF A00A           STX     $A00A
     03C6       TAPEI3 EQU     *
03C6 84 7F             AND A   #$7F
03C8 39                RTS
03C9 0000       TX     FDB     0
                ********************
                * OUTPUT A-REG TO
                * AC-30 CASSETTE
                *
03CB FF 03C9    TAPEOU STX     TX
03CE F6 010A           LDA B   OPSYS
03D1 D1 01             CMP B   MIKBUG
03D3 27 06             BEQ     TAPEO2
03D5 FE 010D           LDX     TAPE
03D8 FF A00A           STX     $A00A
03DB BD 0118    TAPEO2 JSR     OUTCH
03DE F6 010A           LDA B   OPSYS
03E1 D1 01             CMP B   MIKBUG
03E3 27 06             BEQ     TAPEO3
03E5 FE 010B           LDX     TERM
03E8 FF A00A           STX     $A00A
     03EB       TAPEO3 EQU     *
03EB 39                RTS
                ********************
                * INITALIZE PORT
                * THEN TURN MOTOR
                * ON.
                *
     03EC       TAPEON EQU     *
03EC BD 0412           JSR     TINIT
03EF 81 01             CMP A   #01
03F1 26 03             BNE     TAPEN1
                *
                * TURN READER ON
                *
03F3 86 11             LDA A   #$11
03F5 06         BRA    TAP                EN2
03F6 81 02      TAPEN1 CMP A   #2
03F8 26 05             BNE     TAPEN3
                *
                * TURN PUNCH ON
                *
03FA 86 12             LDA A   #$12
03FC BD 011E    TAPEN2 JSR     OUTAPE
     03FF       TAPEN3 EQU     *
03FF 39                RTS
                ********************
                * TURN MOTOR OFF
                *
     0400       TAPEOF EQU     *
0400 81 01             CMP A   #1
0402 26 04             BNE     TAPEF1
                *
                * TURN READER OFF
```

```
                *
0404 86 13             LDA A  #$13
0406 20 06             BRA    TAPEF2
0408 81 02      TAPEF1 CMP A  #2
040A 26 05             BNE    TAPEF3
                *
                * TURN PUNCH OFF
                *
040C 86 14             LDA A  #$14
040E BD 011E    TAPEF2 JSR    OUTAPE
     0411       TAPEF3 EQU    *
0411 39                RTS
                *
                * INIT TAPE
                * IF NECESSARY...
                *
     0412       TINIT  EQU    *
0412 39                RTS
0413 0020              RMB    32
     0433       END    EQU    *
                       END
BUFFER 0103
```

PROGRAM 3

```
                       NAM    A680B
                       OPT    O,S,L
                ********************
                *
                *      WRITTEN BY
                * BILL TURNER, WB4ALM
                *          AND
                * BILL THAMES, WB4ARN
                *
                * FOR THE ALTAIR
                * 680B SYSTEM WITH
                * KCACR CASSETTE.
                *
                * CONVERTS FROM IAPS
                * TO 'FAST TYPIST'
                * MODE OF OPERATION.
                *
                *     OR
                *
                * WILL DUPLICATE
                * AN IAPFSFORMATTED
                * TAPE
                *
                ********************
                * INCH $FF00, OUTCH $FF81  (TERMINAL I/O)
                * CSTAT $F010, CDATA $F011  (TAPE I/O)
0100                   ORG    $0100
0100 7E 01BB           JMP    START
0103 0400       BUFFER FDB    END
0105 0000       EBUFF  FDB    0
0107 FFFF       DELAY1 FDB    $FFFF
0109 02         DELAY2 FCB    $2
010A 7E FF00    INEEE  JMP    $FF00
010D 7E FF81    OUTEE  JMP    $FF81
0110 7E E1AC    INCH   JMP    $E1AC
0113 7E E1D1    OUTCH  JMP    $E1D1
0116 7E 0394    INTAPE JMP    TAPEIN
0119 7E 03A5    OUTAPE JMP    TAPEOU
011C 30         HCKSUM FCC    /0000/
011D 30
011E 30
011F 30
0120 0000       COUNT  FDB    0
0122 00                FCB    0
0123 0000       SAVEX  FDB    0
0125 0000       SETX1  FDB    0
0127 0000       SETX2  FDB    0
                ** FOLLOWING IS POSTIONAL DEPENDENT: ***
0129 20         MSGNG  FCC    / #/
012A 20
012B 23
012C 2D         NUMBER FCC    /--------/
012D 2D
012E 2D
012F 2D
0130 2D
0131 2D
0132 2D
0133 2D
0134 20                FCC    / IS BAD/
0135 49
0136 53
0137 20
0138 42
0139 41
013A 44
013B 04                FCB    $04
                ** END OF POSTIONAL DEPENDENT AREA. **
013C 0000       NUMBX  FDB    0
     013E       MSGST  EQU    *
013E 0D0A              FDB    $0D0A
0140 20                FCC    / READY TO START? /
0141 52
0142 45
0143 41
0144 44
0145 59
0146 20
0147 54
0148 4F
0149 20
014A 53
014B 54
014C 41
014D 52
014E 54
014F 3F
0150 20
0151 04                FCB    4
     0152       MSGPCH EQU    *
0152 0D0A              FDB    $0D0A
0154 20                FCC    / READY TO RECORD? /
0155 52
0156 45
0157 41
0158 44
0159 59
015A 20
015B 54
015C 4F
015D 20
015E 52
015F 45
0160 43
0161 4F
0162 52
0163 44
0164 3F
0165 20
0166 04                FCB    4
     0167       MSGDUP EQU    *
0167 0D0A              FDB    $0D0A
0169 20                FCC    / TYPE 'D' TO DUPLICATE /
016A 54
016B 59
016C 50
016D 45
016E 20
016F 22
0170 44
0171 22
0172 20
0173 54
0174 4F
0175 20
0176 44
0177 55
0178 50
0179 4C
017A 49
017B 43
017C 41
017D 54
017E 45
017F 20
0180 0D0A              FDB    $0D0A
0182 20                FCC    / OR 'C' TO CONVERT :/
0183 4F
0184 52
0185 20
0186 22
0187 43
0188 22
0189 20
018A 54
018B 4F
018C 20
018D 43
018E 4F
018F 4E
0190 56
0191 45
0192 52
0193 54
0194 20
0195 3A
0196 04                FCB    $04
     0197       MSGF   EQU    *
0197 0D0A              FDB    $0D0A
0199 20                FCC    / ALL DONE --- READY TO CONTINUE? /
019A 41
019B 4C
019C 4C
019D 20
019E 44
019F 4F
01A0 4E
01A1 45
01A2 20
01A3 2D
01A4 2D
01A5 2D
01A6 20
01A7 52
01A8 45
01A9 41
01AA 44
01AB 59
01AC 20
01AD 54
01AE 4F
01AF 20
01B0 43
01B1 4F
01B2 4E
01B3 54
01B4 49
01B5 4E
01B6 55
01B7 45
01B8 3F
01B9 20
01BA 04                FCB    04
                *
                * INIT STACK POINTER
                * DO DUMMY READ
                * FROM KCACR INTERFACE,
                * ISSUE 'INPUT?' MSG,
                * WAIT FOR -CR-
                *
01BB 8E 03FF    START  LDS    #END-1
01BE CE 013E           LDX    #MSGST
01C1 BD 039D           JSR    TI2
01C4 BD 0372           JSR    PDATA1
01C7 BD 036A           JSR    WAIT
                *
                * TURN TAPE ON
                *
```

```
01CA FE 0103          LDX     BUFFER
01CD BD 03B3          JSR     TAPEON
                *
                * GET CHAR, LOOP
                * UNTIL -SOH-
                * FOUND.IF -EOT-
                * ASSUME END-OF-FILE.
                *
01D0 BD 0116    GET1   JSR     INTAPE
01D3 81 01             CMP A   #$01
01D5 26 F9             BNE     GET1
01D7 20 0B             BRA     GOTSOH
01D9 BD 0116    GETSOH JSR     INTAPE
01DC 81 04             CMP A   #$04
01DE 27 2B             BEQ     GOTEOT
01E0 81 01             CMP A   #$01
01E2 26 F5             BNE     GETSOH
                *
                * WE GOT A -SOH-
                * RESET CHECKSUM COUNT
                * CLEAR BLOCK-ID FIELD.
                *
01E4 A7 00      GOTSOH STA A   0,X
01E6 08                INX
01E7 7F 0120           CLR     COUNT
01EA 7F 0121           CLR     COUNT+1
01ED FF 0123           STX     SAVEX
01F0 CE 012C           LDX     #NUMBER
01F3 FF 013C           STX     NUMBX
01F6 CE 2020           LDX     #$2020
01F9 FF 012C           STX     NUMBER
01FC FF 012E           STX     NUMBER+2
01FF FF 0130           STX     NUMBER+4
0202 FF 0132           STX     NUMBER+6
0205 FE 0123           LDX     SAVEX
0208 BD 0116    GET    JSR     INTAPE
                *
                * SAVE ALL CHARS IN
                * BUFFER.
                *
020B A7 00      GOTEOT STA A   0,X
020D 08                INX
                *
                * ADD CHAR TO CHECKSUM
                *
020E 36                PSH A
020F BB 0121           ADD A   COUNT+1
0212 B7 0121           STA A   COUNT+1
0215 24 03             BCC     CKEOT
0217 7C 0120           INC     COUNT
021A 32         CKEOT  PUL A
                *
                * WAS CHAR AN -EOT-
                *
021B 81 04             CMP A   #$04
021D 26 03             BNE     CKSTX
021F 7E 02C4           JMP     ENDIN
                *
                * WAS CHAR AN -STX-?
                *
0222 81 02      CKSTX  CMP A   #$02
0224 26 0C             BNE     CKETX
0226 FF 0123           STX     SAVEX
                *
                * -STX- FOUND, DON'T
                * SAVE ANY MORE
                * BLOCK-ID CHARS
                *
0229 CE 0000           LDX     #0
022C FF 013C           STX     NUMBX
022F 7E 02A1           JMP     SNUMB1
                *
                * -ETX-? IF SO
                * CALCULATE CHECKSUM
                * AND COMPARE TP
                * TRANSMITTED VALUE.
                *
0232 81 03      CKETX  CMP A   #$03
0234 26 68             BNE     SNUMB
0236 FF 0125           STX     SETX1
                *
                * SAVE XMITTED VALUE.
                *
0239 BD 0116           JSR     INTAPE
023C A7 00             STA A   0,X
023E 08                INX
023F BD 0116           JSR     INTAPE
0242 A7 00             STA A   0,X
0244 08                INX
0245 FF 0127           STX     SETX2
0248 BD 0116           JSR     INTAPE
024B A7 00             STA A   0,X
024D 08                INX
024E BD 0116           JSR     INTAPE
0251 A7 00             STA A   0,X
0253 08                INX
                *
                * PRODUCE 2'S
                * COMPLEMENT, AND
                * AND CONVERT TO
                * 'ASCII' REPRESENTATION.
                *
0254 F6 0121           LDA B   COUNT+1
0257 B6 0120           LDA A   COUNT
025A 50                NEG B
025B 24 01             BCC     CKETX1
025D 4C                INC A
025E 40         CKETX1 NEG A
025F F7 0122           STA B   COUNT+2
0262 FF 0123           STX     SAVEX
0265 CE 011C           LDX     #HCKSUM
0268 BD 0357           JSR     OPACK
026B B6 0122           LDA A   COUNT+2
026E CE 011E           LDX     #HCKSUM+2
0271 BD 0357           JSR     OPACK
                *
                * COM@E@T NOW
                *
0274 CE 011C           LDX     #HCKSUM
0277 A6 00             LDA A   0,X
0279 E6 01             LDA B   1,X
027B FE 0125           LDX     SETX1
027E A1 00             CMP A   0,X
0280 26 33             BNE     NOGOOD
0280 26 33             BNE     NOGOOD
0282 E1 01             CMP B   1,X
0284 26 2F             BNE     NOGOOD
0286 CE 011E           LDX     #HCKSUM+2
0289 A6 00             LDA A   0,X
028B E6 01             LDA B   1,X
028D FE 0127           LDX     SETX2
0290 A1 00             CMP A   0,X
0292 26 21             BNE     NOGOOD
0294 E1 01             CMP B   1,X
0296 26 1D             BNE     NOGOOD
0298 FE 0123           LDX     SAVEX
                *
                * THIS BLOCK WAS OK
                * GO GET ANOTHER...
                *
029B 7E 01D9           JMP     GETSOH
029E FF 0123    SNUMB  STX     SAVEX
02A1 FE 013C    SNUMB1 LDX     NUMBX
                *
                * ARE WE SAVING THIS CHAR
                * AS PART OF THE
                * BLOCK-ID?
                *
02A4 8C 0000           CPX     #0
02A7 27 06             BEQ     ENUMB
02A9 A7 00             STA A   0,X
02AB 08                INX
02AC FF 013C           STX     NUMBX
02AF FE 0123    ENUMB  LDX     SAVEX
02B2 7E 0208           JMP     GET
                *
                * BLOCK WAS NO GOOD
                * ISSUE MESSAGE.
                *
     02B5       NOGOOD EQU     *
02B5 FF 0123           STX     SAVEX
02B8 CE 0129           LDX     #MSGNG
02BB BD 0372           JSR     PDATA1
02BE FE 0123           LDX     SAVEX
02C1 7E 0208           JMP     GET
                ********************
                * OUTPUT BUFFER:   *
                ********************
     02C4       ENDIN  EQU     *
02C4 FF 0105           STX     EBUFF
                *
                * TURN READER OFF
                * AND WAIT FOR OK
                * TO PROCEED
                *
02C7 BD 03C4           JSR     TAPEOF
02CA CE 0152           LDX     #MSGPCH
02CD BD 0372           JSR     PDATA1
02D0 BD 036A           JSR     WAIT
                *
                * OPTION: 'D' OR 'C'?
                *
02D3 CE 0167    OPTION LDX     #MSGDUP
02D6 74 0372           LSR     PDATA1
02D9 BD 010A           JSR     INEEE
02DC D1 44             CMP B   'D
02DE 27 4D             BEQ     PUT5
02E0 D1 43             CMP B   'C
02E2 27 02             BEQ     PUT0
02E4 20 ED             BRA     OPTION
                *
                * CONVERSION OPTION
                * SELECTED...
                * TURN PUNCH ON
                * CREATE LEADER
                *
     02E6       PUT0   EQU     *
02E6 BD 03B3           JSR     TAPEON
02E9 BD 037F           JSR     DELAYS
02EC BD 037F           JSR     DELAYS
02EF BD 037F           JSR     DELAYS
02F2 FE 0103           LDX     BUFFER
                *
                * OUTPUT EACH DATA LINE
                * WITH A TIME DELAY
                * AFTER THE -CR-
                * STOP WHEN -EOT-
                * FOUND
                *
                **********
                * SKIP ALL DATA UP TO
                * AND INCLUDING THE
                * -STX-
                *
02F5 A6 00      PUT    LDA A   0,X
02F7 08                INX
02F8 81 04             CMP A   #$04
02FA 27 4C             BEQ     DONE
02FC 81 02             CMP A   #$02
02FE 26 F5             BNE     PUT
                *
                * OUTPUT ALL CHARS
                * UNTIL -ETX-
                *
                * DON'T FORGET THE
                * -DLE- SEQUENCES.
                *
                * EVERYTHING BETWEEN
                * -STX- AND -ETX- IS
                * PART OF THE BASIC
                * STATEMENT.
                *
0300 A6 00      PUT1   LDA A   0,X
0302 08                INX
                *
                * -EOT- CHAR?
                *
0303 81 04             CMP A   #$04
0305 27 1C             BEQ     PUT3
                *
                * -ETX- CHAR?
                *
0307 81 03             CMP A   #$03
0309 27 0E             BEQ     PUT2
                *
                * -DLE- CHAR?
     @                 *
030B 81 10             CMP A   #$10
030D 26 05             BNE     PUT11
                *
                * WAS -DLE- ...
                * CONVERT NEXT CHAR
                * BACK TO ORIG.VALUE
                *
030F A6 00             LDA A   0,X
0311 08                INX
0312 84 3F             AND A   #$3F
                *
                * OUTPUT CHAR
                * IN A-REG
                *
0314 BD 0119    PT11   JSR     OUTAPE
0317 20 E7             BRA     PUT1
                *
                * -ETX- WAS FOUND
                * OUTPUT A -CR-
                * INSTEAD. THEN
                * PAUSE (ALLOWS
                * RECEIVING SYSTEM
                * TO ECHO A -LF-
                * WITHOUT INTERFERING
                * WITH INCOMING DATA.
                *
0319 86 0D      PUT2   LDA A   #$0D
031B BD 0119           JSR     OUTAPE
031E BD 037F           JSR     DELAYS
0321 20 D2             BRA     PUT
                *
                * -EOT- WAS FOUND
                * OUTPUT FINAL -CR-
                *
0323 86 0D      PUT3   LDA A   #$0D
0325 BD 0119           JSR     OUTAPE
0328 BD 037F           JSR     DELAYS
032B 20 1B             BRA     DONE
                *
                * DUPLICATION OPTION
                * SELECTED...
                *
                * TURN PUNCH ON
                * CREATE LEADER
                *
032D BD 03B3    PUT5   JSR     TAPEON
0330 BD 037F           JSR     DELAYS
0333 BD 037F           JSR     DELAYS
0336 BD 037F           JSR     DELAYS
0339 FE 0103           LDX     BUFFER
                *
                * OUTPUT ENTIRE BUFFER
                * TO TAPE --- STOP WHEN
                * -EOT- CHAR FOUND.
                *
033C A6 00      PUT6   LDA A   0,X
033E 08                INX
033F BD 0119           JSR     OUTAPE
0342 81 04             CMP A   #$04
0344 27 02             BEQ     DONE
0346 20 F4             BRA     PUT6
                *
                * TURN PUNCH OFF
                * WAIT FOR OK
                * TO PROCEED
                *
     0348       DONE   EQU     *
0348 BD 03C4           JSR     TAPEOF
034B CE 0197           LDX     #MSGF
034E BD 0372           JSR     PDATA1
0351 BD 036A           JSR     WAIT
0354 7E 01BB           JMP     START
                ********************
                * SUBROUTINES:     *
                ********************
                *
                * CONVERT BYTE IN
                * A-REG TO A 2-BYTE
                * ASCII SEQUENCE OF
                * HEX '40' THRU '4F'
                *
0357 36         OPACK  PSH A
0358 44                LSR A
0359 44                LSR A
035A 44                LSR A
035B 44                LSR A
035C 84 0F             AND A   #$0F
035E 8A 40             ORA A   #$40
0360 A7 00             STA A   0,X
0362 32                PUL A
0363 84 0F             AND A   #$0F
0365 8A 40             ORA A   #$40
0367 A7 01             STA A   1,X
0369 39                RTS
                ********************
                * WAIT FOR -CR- REPLY
                *
     036A       WAIT   EQU     *
                *  JSR INCH
036A BD 010A           JSR     INEEE
036D C1 0D             CMP B   #$0D
036F 26 F9             BNE     WAIT
0371 39                RTS
```

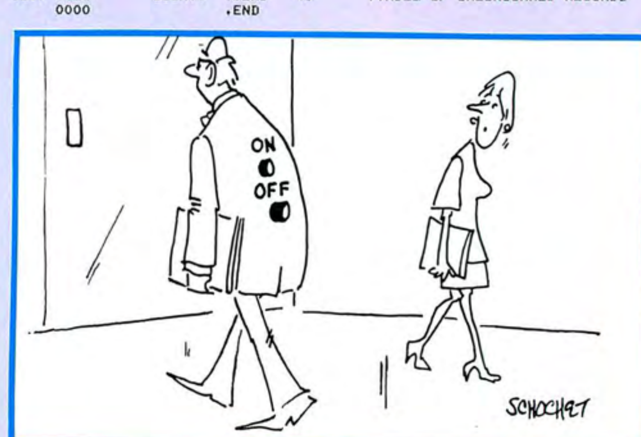
```
                   ********************
                   * OUTPUT ALL CHARS
                   * OF A MESSAGE UP
                   * TO BUT NOT INCLUDING
                   * A HEX '04' CHAR.
                   * X-REG POINTS TO
                   * START OF MESSAGE
                   * UPON ENTRY.
                   *
0372 E6 00    PDATA1 LDA B  0,X
0374 C1 04           CMP B  #$04
0376 27 06           BEQ    PDATA2
              *  JSR OUTCH
0378 BD 010D         JSR    OUTEE
037B 08              INX
037C 20 F4           BRA    PDATA1
037E 39       PDATA2 RTS
                   ********************
                   * SOFTWARE TIMING
                   * DELAY LOOP.
                   * CHANGE VALUES OF
                   * DELAY1 AND DELAY2
                   * TO CHANGE LENGTH
                   * OF DELAY
                   *
037F FF 0123  DELAYS STX    SAVEX
0382 FE 0107         LDX    DELAY1
0385 F6 0109         LDA B  DELAY2
0388 5A       DEL1   DEC B
0389 27 05           BEQ    DEL3
038B 09       DEL2   DEX
038C 26 FD           BNE    DEL2
038E 20 F8           BRA    DEL1
0390 FE 0123  DEL3   LDX    SAVEX
0393 39              RTS
                   ********************
                   * GET A CHAR FROM
                   * KCACR INTERFACE
                   * INTO THE A-REG.
                   *
0394 FF 03A3  TAPEIN STX    TX
0397 B6 F010  TI     LDA A  $F010
039A 46              ROR A
039B 25 FA           BCS    TI
039D B6 F011  TI2    LDA A  $F011
03A0 84 7F           AND A  #$7F
03A2 39              RTS
03A3 0000     TX     FDB    0
                   ********************
                   * OUTPUT A-REG TO
                   * KCACR INTERFACE
                   *
03A5 FF 03A3  TAPEOU STX    TX
03A8 36              PSH A
03A9 B6 F010  TO     LDA A  $F010
03AC 2B FB           BMI    TO
03AE 32              PUL A
03AF B7 F011         STA A  $F011
03B2 39              RTS
                   ********************
                   * CLEAR OUT XMIT
                   * UART BY SENDING
                   * THREE NULLS, AND
                   * THEN TURN MOTOR
                   * ON.
                   *
     03B3     TAPEON EQU    *
03B3 86 00           LDA A  #$00
03B5 BD 0119         JSR    OUTAPE
03B8 BD 0119         JSR    OUTAPE
03BB BD 0119         JSR    OUTAPE
03BE 86 7F           LDA A  #$7F
03C0 B7 F010         STA A  $F010
03C3 39              RTS
                   ********************
                   * TURN MOTOR OFF
                   *
     03C4     TAPEOF EQU    *
03C4 86 BF           LDA A  #$BF
03C6 B7 F010         STA A  $F010
03C9 39              RTS
0400                 ORG    $0400
     0400     END    EQU    *
                     END
BUFFER 0103

EBUFF  0105

DELAY1 0107

DELAY2 0109

INEEE  010A

OUTEE  010D

INCH   0110
```

PROGRAM 4

```
        ; PROGRAM TO LOAD AND DUMP INTERFACE AGE
        ; PUBLICATION FORMAT FOR AN 8080 MICROPROCESSOR
        ;
        ; MARCH 14,1978
        ;
        ; WRITTEN BY ALAN R. MILLER
        ; NEW MEXICO TECH, SOCORRO, NM  87801
        ;
        ;
8000    .PHASE  $8000
        ;
F3A0    STACK   ==      $F3A0
F800    MONIT   ==      $F800   ;GO TO MONITOR ON G
0012    FSTAT   ==      $12     ;FILE SOURCE STATUS
0013    FDATA   ==      $13     ;FILE SOURCE DATA
0001    FIMSK   ==      1       ;INPUT MASK
0002    FOMSK   ==      2       ;OUTPUT MASK
0012    TSTAT   ==      $12     ;TAPE STATUS
```

```
        0013          TDATA    ==      $13      ;TAPE DATA
        0002          TOMSK    ==      2        ;TAPE-OUTPUT MASK
        0010          CSTAT    ==      $10      ;CONSOLE STATUS
        0011          CDATA    ==      $11      ;CONSOLE DATA
        0001          CIMSK    ==      1        ;CONSOLE INPUT MASK
        0002          COMSK    ==      2        ;CONSOLE OUTPUT MASK
        0001          SOH      ==      1        ;START OF BLOCK
        0002          STX      ==      2        ;START OF TEXT
        0003          ETX      ==      3        ;END OF TEXT
        0004          EOT      ==      4        ;END OF FILE
        0010          DLE      ==      16       ;CONTROL CHARACTER FOLLOWS
        000D          CR       ==      13       ;CARRIAGE RETURN
        000A          LF       ==      10       ;LINE FEED
                      ;
8000    31 A0F3       START:   LXI     SP,STACK
8003    AF                     XRA     A        ;GET A ZERO
8004    32 EB81                STA     LFLAG    ;RESET LOAD FLAG
8007    32 EA81                STA     EFLAG    ;ZERO ERROR COUNT
800A    CD CC81                CALL    CRLF
800D    3E 3E                  MVI     A,'>'
800F    CD 6381                CALL    OUTT     ;PRINT A PROMPT
8012    CD 4E81                CALL    READ     ;INPUT TASK FROM CONSOLE
8015    FE 44                  CPI     'D'      ;DUMP FILE TO TAPE
8017    CA 1E81                JZ      DUMP
801A    FE 4C                  CPI     'L'      ;LOAD TO MEMORY
801C    CA 2980                JZ      LOAD
801F    FE 43                  CPI     'C'      ;COPY ALL TO MEMORY
8021    C2 9D81                JNZ     ERROR
8024    3E 01                  MVI     A,1
8026    32 EB81                STA     LFLAG    ;SET LOAD FLAG
                      ;
                      ; INPUT FILE FROM TERMINAL AND LOAD INTO MEMORY
                      ; [H,L] IS THE MEMORY POINTER, [B,C] IS THE DATA SUM
                      ;
8029    CD 6F81       LOAD:    CALL    READHL   ;INPUT START ADDRESS
802C    CD C481                CALL    GO       ;WAIT FOR CARRIAGE RETURN
802F    EB                     XCHG
8030    21 0782                LXI     H,CSUMT  ;START OF CHECKSUM TABLE
8033    22 F481                SHLD    CPNTR    ;RESET POINTER TO START
8036    EB                     XCHG
8037    CD D080       LOADN:   CALL    INBYTE   ;GET A BYTE
803A    FE 04                  CPI     EOT      ;END OF FILE?
803C    CA EF80                JZ      DONE     ;YES
803F    FE 01                  CPI     SOH      ;START OF BLOCK?
8041    C2 3780                JNZ     LOADN    ;LOOP UNTIL START
8044    AF                     XRA     A        ;GET A ZERO
8045    47                     MOV     B,A      ;ZERO THE DATA SUM
8046    4F                     MOV     C,A
8047    11 EC81                LXI     D,BLOCK  ;CURRENT BLOCK NUMBER
                      ;
                      ; INPUT THE BLOCK NUMBER (ONE-EIGHT HEX CHARACTERS)
                      ;
804A    D5            BLOCKN:  PUSH    D
804B    CD D080                CALL    INBYTE   ;GET A BYTE
804E    D1                     POP     `D
804F    12                     STAX    D        ;SAVE DIGIT OF BLOCK NUMBER
8050    13                     INX     D
8051    FE 02  `              CPI     STX      ; END OF NUMBER?
8053    C2 4A80                JNZ     BLOCKN   ;LOOP UNTIL END
8056    1B                     DCX     D        ;BACK POINTER TO STX
8057    3E 20                  MVI     A,' '
8059    12 `                   STAX    D        ;CLEAR STX
          `           ; INPUT THE TEXT
                      ;
805A    CD D080       TEXT:    CALL    INBYTE   ;GET A BYTE
0'C*'  FE 0#      @      API      ETX      ;END OF BLOBJ
                                                          805F   CA 7C80
8062    FE 10                  CPI     DLE      ;CONTROL CHARACTER?
8064    C2 6F80                JNZ     MOVE     ;NO, PUT BYTE IN MEMORY
8067    C@ E380                CALL    BYTE     ;G#HE CONTROL CHARACTER
886A    CD D380                CALL    INBY2    ;ADD TO CHECK3_M
806D    E6 1F                  ANI     $1F      ;CONVERT TO CONTROL CHARACTER
806F    5F            MOVE>    MOV     E,A
8070    3A 5(            LDA cFLAG    ;SEE IF LOADING ALL
8073    B7  `                  ORA     A
80w4    C2 MPSNZ        TEXT       ;SKIP SECOND PRINTING
8077    73                     MOV     M,E      ;PUT BYTE IN MEMORY
8078    23                     Pr
                                 B `        ;INCREMENT MEMORY POINTER
8079    C3 5A80                JMP     TEXT     ;NEXT BYTE
                      ;
        C+ND OF BOCK.  GET CHECKSUM AND COMPARE TO
                      ; CALCULATED SUM.
                      ;
807C    3A EB81       DEND:    LDA     LFLAG    ;CHECK LOAD FLAG
P 807F   B7           ORA      A             ;SEE IF ZERO
8080    C2 8680                JNZ     DEN2     ;SKIP IF FULL LOAD
8083    36 0D                  MVI     M,CR     ;PUT CARRIATETURN IN MEMORY
8085    23                     INX     H        ;INCREMENT POINTER
8086    C5            DEN2:    PUSH    B        ;SAVE SUM
8087    CD B980                CALL    FIXSM    ;GET FIRST TWO CHARACTERS
808A    57                     MOV     D,A      ;PUT IN H
808B    CD B980                CALL    FIXSM    ;GET SECOND TWO CHARACTERS
808E    5F                     MOV     E,A      ;PUT IN L
808F    C1                     POP     `        ;RETRIEVE SUM
8090    EB                     XCHG             ;c*CKSUM TO H,L
CJ
  J              DAD      B          ;ADD DATA SUM TO CHECKSUM
8092    7C                     MOV     A,H
8093    B5                     ORA     L        ;SEE IF BOTH H AND L ARE ZERO
8094    EB                     XCHG             ;RESTORE POINTER TO H,L
8095    CA 3780                JZ      LOADN    ;OK, START NEXT BLOCK
                      ;
                      ; CHECKSUM ERROR.  SAVE BLOCK NUMBER IN BUFFER FOR
                      ; LATER LISTING.  (THERE WON'T BE TIME NOW).
                      ;
8098    E5                     PUSH    H
8099    11 EC81                LXI     D,BLOCK  ;BLOCK NUMBER
809C    2A F481                LHLD    CPNTR    ;POINTER TO CHECKSUM TABLE
809F    1A            NCHAR:   LDAX    D        ;GET BLOCK CHARACTER
80A0    77                     MOV     M,A      ;PUT IN TABLE
80A1    13                     INX     D
80A2    23                     INX     H
80A3    22 F481                SHLD    CPNTR    ;SAVE POINTER
80A6    FE 20                  CPI     ' '
80A8    C2 9F80                JNZ     NCHAR    ;NEXT CHARACTER
80AB    36 01                  MVI     M,1      ;PUT BINARY 1 AT END OF TABLE
80AD    3A EA81                LDA     EFLAG    ;FTECH ERROR COUNT
80B0    3C                     INR     A        ;INCREMENT IT
80B1    27                     DAA              ;CONVERT TO DECIMAL
80B2    32 EA81                STA     EFLAG    ;SAVE NEW VALUE
80B5    E1                     POP     H        ;RESTORE TEXT POINTER
80B6    C3 3780                JMP     LOADN
                      ;
                      ; INPUT TWO BYTES FOR HALF OF CHECKSUM.
                      ; CONVERT TO ONE BINARY BYTE IN A
                      ;
80B9    CD C680       FIXSM:   CALL    BYTES    ;GET FIRST BYTE
80BC    07                     RLC
```

```
80BD  07              RLC             ;ROTATE TO UPPER HALF
80BE  07              RLC
80BF  07              RLC
80C0  47              MOV     B,A     ;SAVE IN L
80C1  CD C680         CALL    BYTES   ;GET SECOND PART
80C4  B0              ORA     B       ;COMBINE BOTH PARTS
80C5  C9              RET
                ;
                ; GET A BYTE WITHOUT ALTERING CHECKSUM
                ;
80C6  CD E380   BYTES: CALL   BYTE
80C9  5F              MOV     E,A
80CA  CD DA80         CALL    INBY3   ;SEE IF LOADING ALL
80CD  E6 0F           ANI     $0F     ;KEEP LOWER THREE BITS
80CF  C9              RET
                ;
                ; INPUT A BYTE FROM FILE, ADD TO CHECKSUM.
                ; CHECK FOR END OF FILE.
                ;
80D0  CD E380   INBYTE: CALL  BYTE    ;GET THE BYTE
80D3  5F        INBY2: MOV    E,A     ;SAVE BYTE IN E
80D4  81              ADD     C       ;ADD !O SUM
80D5  4F              MOV     C,A     ;PUT NEW SUM BACK TO C
80D6  D2 DA80         JNC     INBY3   ;DONE IF NO CARRY
80D9  04              INR     B       ;ADD CARRY TO B
80DA  3A EB81   INBY3: LDA    LFLAG   ;CHECK LOAD FLAG
80DD  B7              ORA     A
80DE  7B              MOV     A,E
80DF  C8              RZ
80E0  77              MOV     M,A     ;PUT CONTROL BYTES IN MEMORY
80E1  23              INX     H       ;INCREMENT POINTER
80E2  C9              RET
                ;
                ; INPUT A BYTE FROM FILE TERMINAL
                ;
80E3  DB 12     BYTE:  IN     FSTAT   ;CHECK FILE STATUS
80E5  E6 01           ANI     FIMSK   ;MASK FOR INPUT
80E7  CA E380         JZ      BYTE    ;LOOP UNTIL READY
80EA  DB 13           IN      FDATA   ;GET BYTE
80EC  E6 7F           ANI     127     ;STRIP PARITY
80EE  C9              RET
                ;
                ; END OF FILE.  PRINT POINTER.
                ;
80EF  CD A581   DONE:  CALL   OUTHL
80F2  36 01           MVI     M,1     ;BINARY 1 MARKS END
80F4  3A EA81         LDA     EFLAG   ;SEE IF ANY CHECKSUM ERRORS
80F7  B7              ORA     A
80F8  CA 0080         JZ      START   ;NO, RESTART
80FB  47              MOV     B,A     ;SAVE ERROR COUNT
80FC  CD CC81         CALL@P  CRLF
80FF  78              MOV     A,B
8100  CD AA81         CALL    OUTHX   ;PRINT NUMBER OF ERRORS
8103  21 F6cXI     H,EMESG ;POINT TO ERROR MESSAGE
8106  CD E081         CALL    SENDM   ;PRINT IT
                ;
                ; PRINT BLOCK NUMBERS WHICH HAD CHECKSUM ERRORS
                ;
8109  CD CC81   PLINES: CALL  CRLF
810C  7E        PLIN2: MOV    A,M     ;GET CHARACTER
810D  23              INX     H       ;INCREMENT POINTER
810E  FE 01           CPI     1       ;BINARY 1 AT END OF TABLE
8110  CA 0080         JZ      START   ;DONE
8113  FE 20           CPI     ' '     ;BLANK AT END OF NUMBER
811=  CA 0981         JZ      PLINES  ;START ON NEXT NUMBER
8118`  CD 6381        CALL    OUTT    ;PRINT CHATER
811B  C3 0C81         JMP     PLIN2   ;NEXT CHARACTER
                ;
                ; ROUTI* TO SEND TEXT IN MEMORY TO TAPE PORT.
                ; ADD TIME DELAY AFTER CARRIAGE RETURN.
                ;
811E  CD 6F81   DUMP:  CALL   READHL  ;GET STARTING ADDRESS
8121  CD C481         CALL    GO      ;WAIT FOR CARRIAGE RETURN
8124  7E        DMP2:  MOV    A,M     ;FETCH BYTE
8125  23              INX     H       ;INCREMENT POINTER
8126  FE 01           CPI     1       ;BINARY 1 AT BUFFER END
8128  CA 0080         JZ      START   ;DONE
812B  CD 4281         CALL    TOUT    ;OUTPUT BYTE
812E  FE 0D           CPI     CR      ;CHECK FOR CARRIAGE RETURN
8130  C2 2481         JNZ     DMP2    ;CONTINUE IF NOT
8133  16 78           MVI     D,120   ;OUTER TIMING LOOP
8135  1E C8     DMP3:  MVI    E,200   ;INNER TIMING LOOP
8137  1D        DMP4:  DCR    E
8138  C2 3781         JNZ     DMP4    ;LOOP ON E
813B  15              DCR     D
813C  C2 3581         JNZ     DMP3    ;LOOP OF D
813F  C3 2481         JMP     DMP2    ;DELAY DONE
                ;
                ; ROUTINE TO OUTPUT A BYTE TO TAPE
                ;
8942  F5        TOUT:  PUSH   PSW
8143  DB 12     TOUTW: IN     TSTAT   ;CHECK STATUS
8145  E6 02           ANI     TOMSK   ;MASK UNWANTED BITS
8147  CA 4381         JZ      TOUTW   ;LOOP UNTIL READY
814A  F1              POP     PSW     ;GET BYTE
814B  D3 13           OUT     TDATA
814D  C9              RET
                ;
                ; CONSOLE-INPUT ROUTINE
                ;
814E  DB 10     READ:  IN     CSTAT   ;CHECK STATUS
8150  E6 01           ANI     CIMSK   ;MASK FOR INPUT
8152  CA 4E81         JZ      READ    ;LOOP UNTIL READY
8155  DB 11           IN      CDATA   ;GET DATA
8157  E6 7F           ANI     127     ;STRIP PARITY
8159  FE 03           CPI     3       ;CONTRIL-C
815B  CA 0080         JZ      START   ;RESTART
815E  FE 18           CPI     24      ;CONTROL-X
8160  CA 00F8         JZ      MONIT   ;RETURN TO MONITOR
                ;
                ; CONSOLE OUTPUT ROUTINE
                ;
8163  F5        OUTT:  PUSH   PSW
8164  DB 10     OUTW:  IN     CSTAT   ;CHECK STATUS
816&  E6 02           ANI     COMSK   ;MASK FOR OUTPUt
8168  CA 6481         JZ      OUTW    ;LOOP UNTIL READY
816B  F1              POP     PSW
816C  D3 11           OUT     CDATA
816E  C9              RET
                ;
                ; INPUT AN ADDRESS TO H,L FROM THE CONSOLE
                ;
816F  D5        READHL: PUSH  D
8170  CD 7A81         CALL    RDHEX   ;INPUT HIGH HALF
8173  67              MOV     H,A
8174  CD 7A81         CALL    RDHEX   ;INPUT LOW HALF
8177  6DPk=V  B  L,A
8178  D1              POP     D
8179  C9              RET
                ;
                ; INPUT TWO HEX CHARACTERS AND CONVERT TO
                ; A BINARY BYTE IN E
                ;
817A  CD 8881   RDHEX: CALL   HEX2    ;READ UPPER CHARAcTER
817D  07              RLC
817E  17              RAL             ;ROTATE
817F  17              RAL             ;TO UPPER HALF
8180  17              RAL
8181  5F              MOV     E,A
8182  CD 8881         CALL    HEX2    ;READ LOWER HALF
8185  83              ADD     E       ;COMBINE BOTH
8186  5F              MOV     E,A     ;SAVE IN E
8187  C9              RET
                ;
                ; INPUT A HEX CHARACTER TO A
                ;
8188  CD 4E81   HEX2:  CALL   READ    ;CONSOLE INPUT
818B  D6 30           SUI     '0'     ;REMOVE ASCII BIAS
818D  DA 9D81         JC      ERROR   ;ERROR, LESS THAN '0'
8190  FE 17           CPI     23
8192  D2 9D81         JNC     ERROR   ;ERROR, BREATER THAN 'F'
8195  FE 0A           CPI     10
8197  D8              RC              ;A NUMBER 0-9
8198  D6 07           SUI     7
819A  FE 0A           CPI     10
819C  D0              RNC             ;A LETTER A-F
819D  3E 3F     ERROR: MVI    A,'?'
819F  CD 6381         CALL    OUTT    ;PRINT ? FOR ERROR
81A2  C3 0080         JMP     START
                ;
[<UTPUT A DOUBLE BYTE IN HEX
                ;
81A5  7C        OUTHL: MOV    A,H     ;GET H
81A6  CD AA81         CALL    OUTHX   ;PRINT IT
81A9  7D              MOV     A,L     ;GET L
                ;
                ; CONVERT A BINARY BYTE TO TWO HEX CHARACTERS
                ; AND PRINT THEM
                ;
81AA  F5        OUTHX: PUSH   PSW
81AB  F5              PUSH    PSW
81AC  1F              RAR
81AD  1F              RAR             ;ROTATE UPPER
81AE  1F              RAR             ;CHARACTER TO
81AF  1F              RAR             ;LOWER
81B0  CD B981         CALL    HEX1    ;OUTPUT UPPER CHARACTER
81B3  F1              POP     PSW     ;OUTPUT LOWER CHARACTER
81B4  CD B981         CALL    HEX1
81B7  F1              POP     PSW
81B8  C9              RET
0               ;
                ; OUTPUT A HEX CHARACTER FROM
                ; LOWER FOUR BITS
                ;
81B9  E6 0F     HEX1:  ANI    $0F     ;MASK UPPER FOUR BITS
81BB  C6 90           ADI     $90
81BD  27              DAA             ;INTER DAA TRICK
81BE  CE 40           ACI     $40
81C0  27              DAA
81C1  C3 6381         JMP     OUTT
                ;
                ; LOOK FOR A CAR. RETURN AT END OF CONSOLE INPUT LINE
                ;
81C4  CD 4E81   GO:    CALL   READ
81C7  FE 0D           CPI     CR
81C9  C2 9D81         JNZ     ERROR
                ;
                ; CARRIAGE RETURN, LINE FEED AND NULLS
                ;
81CC  3E 0D     CRLF:  MVI    A,CR
81CE  CD 6381         CALL    OUTT
81D1  3E 0A           MVI     A,LF
81D3  CD 6381         CALL    OUTT
81D6  AF              XRA     A       ;GET A NULL
81D7  CD 6381         CALL    OUTT
81DA  CD 6381         CALL    OUTT
81DD  C3 6381         JMP     OUTT
`               ;
                ; SEND AN ASCII MESSAGE TO CONSOLE UNTIL A BINARY
                ; ONE IS FOUND.  [H,L] IS MEMORY POINTER.
                ;
81E0  7E        SENDM: MOV    A,M     ;FETCH BYTE
81E1  CD 6381         CALL    OUTT    ;PRINT IT
81E4  23              INX     H       ;INCREMENT POINTER
81E5  B7              ORA     A       ;ZERO AT END
81E6  C2 E081         JNZ     SENDM   ;KEEP GOING
81E9  C9              RET
                ;
81EA  00        EFLAG: .BYTE  0       ;CHECKSUM ERROR COUN\
81EB  00        LFLAG: .BYTE  0       ;LOAD FLAG
81EC  0B1OCK:  .BLKB   8      ;STORE CuRRENT BLOCK NUMBER
81F4  07 82     CPNTR: .ADDR  CSUMT   ;POINTER TO CHECKSUM TABLE
81F6  20 43 48  EMESG: .ASCIZ ' CHECKSUM ERRORS' ;,0
81F9  45 43 4B
81FC  53 55 4D
81FF  20 45 52
8202  52 4F 52
8205  53 00
8207  0028      CSUMT: .BLKB  40      ;TABLE OF CHECKSUMMED RECORDS
      0000             .END
```



An Evaluation Of the AM-100 Computer System

By James W. Kitzmiller
Jorj Baker — Interviewer

HARDWARE

JORJ: Why did you choose the AM-100 for your hardware as opposed to other micros or minis?

JIM: There's several reasons. The most important thing is how much you get for a buck. The AM-100 seems to be one of the best as far as the number of things you can do per dollar that you invest. Its speed is a lot faster than most other computers. Its expansion capability is about the best you'll find. There may be others comparable, but this is among the best. With many computers, the time from when you place an order until you actually get the system can range up to a year and a half for some systems.

JORJ: By then it's almost outdated!

JIM: Yes, that can happen in some cases. This computer (AM-100) is similar to the Digital Equipment Corporation Computer; the PDP-11 computer. It's got instructions which are very, very close to those of the PDP-11, which is a mini.

JORJ: Is the AM-100 easier to program than a microcomputer?

JIM: I would say this is easier to program than most other computers, it just depends on the micro. The AM-100 has some advantages. One of them is that the word length is not held down, like it is for a lot of computers, when you program in BASIC. A word is usually a letter and a number in BASIC: You have A1, Q2 and so forth. On this you can have year-to-date total as a word and you know exactly what it is. You don't get into conceptual problems of symbols and keeping track of what symbol means what.

JORJ: How does the AM-100 cost compare to other mini and microcomputer costs?

JIM: Its cost is comparable to some of the microsystems at the higher end of the cost range, if you were configuring them for businesses. There may be some of them that are cheaper. In fact, I'm sure micros will come at a cheaper price, but expandability isn't as great on those as it is on the AM-100.

JORJ: Is the AM-100 less expensive than a mini?

JIM: A comparable system we found was the IBM System-32 for $100,000 with software and we can do the same operations for $30,000. The programming ease is much, much better in this than it is in a System-32. System-32's use COBOL and RPG, RPG being the most common. RPG stands for Report Program Generator. It's a language and it deals with a great deal of abstraction in order to say anything. BASIC is much easier to use than RPG.

JORJ: How is the AM-100 superior in business applications?

JIM: Something that is important for businesses is data storage and retrieval and the way data is handled. Business computer languages are written so that data storage and retrieval can be done easily, without a lot of statements and without a whole lot of work. By statements, I mean program steps. The AM-100 version of BASIC is written so it handles that quite well. Also, for business purposes you need access to the data. With hard disks on the AM-100 system it's easy to gain access to the data directly. This is because of the way the AM-100 is designed and interfaces to those big disks. The operating system allows you to go directly to a record on the disk just by knowing a piece of data which is on the disk. These are called keys. Let's say you want to find a person named Baker on there and there's only one on the disk. You can have Baker, the last name, be the key and the computer will go directly to that. This is extremely useful for business applications and for data retrieval.

You can expand the AM-100 to have eight or more terminals in operation at one time using the hard disk system. That allows for system expansion with company expansion. The disk storage can be expanded to 1.2 billion characters on the system at one time, by using 4 big disk drives. That covers the hardware aspects of why the AM-100 is a good basis for a business system.

JORJ: How long has the AM-100 been out?

JIM: My guess is since the first half of 1977.

JORJ: How long has Alpha Micro Systems (AM-100 manufacturer) been in existence?

JIM: A little bit longer than that. The AM-100 is their first product.

JORJ: Do you think Alpha Micro Systems will be around next year or the year after that?

JIM: I'm pretty impressed with them for a number of reasons. One reason is the precautions they have taken to make sure their products will stay out. They have an agreement that if they go under, the software they have written and the techniques for producing the hardware will be made available and turned over to some other company in order to support it. The company that makes chips for the AM-100 system goes out of business, the plans for making the chips will be turned over to Alpha Micro Systems so they can still produce their product.

Another thing about the stability of Alpha Micro Systems is their marketing plan. They're going through dealerships and not going off the deep end as far as

marketing. They're letting their products out to become known. To become a dealer for them you have to be financially solvent and technically competent yourself. One thing, you have to purchase in minimum quantities of five, and that will keep out the little guys. That is part of their overall plan. Another thing, I've been in touch with their technical people and have seen their technical plans, what they have planned to develop, and they do come out with that they've promised they'd come out with. For example, three or four months ago they did not have the trignometric functions on their computer yet and they promised that they would have those out. A few weeks later they did have them out. They have an overall plan for expansion which is quite sane.

JORJ: Is the AM-100 applicable for scientific use?
JIM: Yes. It's got a characteristic called "hardware floating-point arithmetic." That means that the calculations, rather than having to go through some programming procedures to be done, are done automatically by the hardware. Therefore for computational purposes it's much, much faster than a lot of computers.

JORJ: How does the AM-100 CPU cost of $1,500 compare with other CPU costs?
JIM: You can get central processing unit based on the 8080 very inexpensively. For the bigger computers it would be quite a bit more than $1,500. I don't know the exact amount. So it's in between the micro and mini price.

The AM-100 is built on the S-100 bus, which is a structure into which you plug various boards and components. It's an industry standard. The S-100 bus is tailored after the first home computer; the Altair computer. There is a wealth of new products that come out that interface with this S-100 bus; like memory boards, interface boards, speech analyzers and just any new invention that comes out will have a version that will plug into the S-100 bus. So the computer is not an oddball kind that ties you into sticking with a narrow range of gear that can be plugged into it.

Another thing to keep in mind is adding more memory. Sixteen thousand characters of memory would cost about $699 to add on DEC, and Jacquard computers run several thousand dollars for 16,000 characters of memory. It's like buying a Ford. Buying parts for a Ford is much cheaper than buying parts for a Cadillac or Rolls Royce. The only thing is, in terms of performance, the AM-100 is closer to the Cadillac stage.

JORJ: What maintenance options does a system purchaser have and what kind of a time lag is there in service?
JIM: There's actually four ways to do it. Least expensive is a 'time and materials' arrangement. If any anytime there's a problem with the machine, you call and have a person come out and fix it for a certain hourly rate, plus cost of materials. A second way is a maintenance contract where the work is done outside of your office, and if there's a problem you take the computer or component to the shop. A third way is a contract where if there is any kind of a problem the service man comes to your place to fix it. The time it takes him to get there would be a function of the terms of the contract, i.e. the higher the amount paid, the faster the service man shows up. The fourth and ultimate method is, with the amount of cost savings on this particular system over, say, an IBM System 32, you could hire a full time guy to sit there and wait for it to go bad!

JORJ: What has been your down-time experience with the AM-100?
JIM: We've had several problems with it. When we first got it the fuse in the disk drive blew and we didn't know it. So, we took it back the same night, changed the fuse and brought it home again. Several days later the fuse blew again and we just changed the fuse. It hasn't blown since that time. About a month ago the interface went a little bit strange so we took that down to the place where we got our maintenance done. They changed the board, and that was fine. One other thing happened, the printer ribbon ran to the very end and for some reason it didn't turn around and go the other way. It caused a fuse to blow. I changed the fuse and got my hands dirty and fixed the ribbon so it would go the other way.

JORJ: How long have you been using this system?
JIM: Since September 1977. We've gone down to the shop twice. Had we known about the fuse problem, we would've been down there just once.

JORJ: How much down-time, in hours and dollars, has it cost you?
JIM: Dollars were covered by our maintenance contract. If we had been on time and materials, it would've been about $15.00. The maintenance contract *we have costs* us $210.00 for 3 months. That's why I say time and materials is better. Our own time consumed in fixing the system or going to get it fixed, counting driving time, has amounted to about 6 hours.

JORJ: You were originally selling a service on a telephone line to an out-of-house computer. Why did your company to drop the on-line service and adopt the in-house service?
JIM: The phone line had a lot of noise on it, which made the accuracy of the data transmission unreliable. When our push-button phones were installed with different incoming telephone numbers, the noise on the phone was so bad we couldn't use the system at all. Another thing was our phone bill. That alone had gone up to over $500.00 a month just to tie into the computer. The computer was 20 miles away which is farther than most people would go over the phone lines. Also, the rental of the terminal and the device to make the terminal talk over the phone line was $125.00 a month. Our computer time was an additional $7 an hour on top of that. Based on a 40 hour/month usage, out computer bill was around $900.00 a month. The hardware of the AM-100 system, the way we recommend it, would lease for around $250.00 a month. For the applications we're doing now, there was nothing you couldn't do on the AM-100 system that you could do on that bigger system.

SOFTWARE

JORJ: Is there a lot of software available for the AM-100?
JIM: Certain systems software is available that comes with the computer. That's the language that allows you to talk to the computer and lets you copy things from one disk to another disk, just things that make it easy to use a computer. They call that the 'systems software.' There's quite a bit of that available. BASIC and LISP are languages available for that. There is a language close to the machine itself that comes with the system. This is similar to the DEC PDP-11 computers, so these languages are widely known by a lot of people.

In terms of the applications software, we're doing that ourselves. We have a general ledger running, an employment agency information retrieval program running, a personal budget analysis program running, and a number of other programs that are not as commonly used are running on our system. The systems software comes with a text editor which can be used for simple word processing applications without too much difficulty. If you get into advanced work processing applications it becomes more difficult to use this particular text editor for that.

As far as software we're working on now; one is an advanced word processor that will handle more sophisticated word processing applications.

Another program is payroll. This will write payroll checks for people, handle the W-2 forms at the end of the year and handle the appropriate reports that an

employer would have to send into city, state and federal governments for payroll purposes. We have also encountered an interest from temporary employment agencies who pay people different rates. We are developing a special payroll program for them.

We are also developing an inventory control program. Many companies have a bunch of items in stock or to be built and they don't know with which priorities to stock things. They maybe overstocked in some areas and understocked in certain other areas. Overstocking causes a loss because it uses up space and ties up cash in items that are just sitting there. Being understocked causes a loss in potential sales. If it's a manufacturing business, the final production of a completed piece of gear can be held up because one item is not in stock.

We are also working on an accounts receivable and billing program which will keep track of who owes you money, what their addresses are, when was the last time they paid you and when was the last time they received something from you. The program will direct the computer to send out invoices to them, deduct from the amount they owe you, when they send you a check, and handle all the things you need to know along those lines.

Our accounts payable program does the same type of thing in the other direction. You keep track of who you owe money to, and how long you've owed it. Checks will be mailed to those people when you direct checks to be printed. You can keep track of the age of the bills you owe, so you can pay the oldest bills first.

JORJ: Do you have an attorney time billing and program?

JIM: Yes, this is a computer system function whereby it will keep track of how much time a given attorney spends doing a given job for a given client. At the end of a time period, say once every week or month, the client can be billed by those attorneys and the total amount owed to the law firm is recorded. You can also keep track of how much each client owed and keep track of individual attorneys to see how much production they are doing.

JORJ: Are these programs you have developed available to AM-100 users from any other source?

JIM: I have heard that some people either had or were working on attorney time and billing. I also heard there is someone in Texas who has a General Ledger program, so these things are available. A person we deal with had an income tax package for the Am-100 directed to tax accountants. When a person comes in to talk to a tax accountant, the tax man can enter the information from the interview directly from a data entry terminal into the computer, and his tax form will be filled out on the spot. That makes it very convenient.

We also are working on a management by objectives program application. I don't know of anyone else who is doing it. It's a project management function and could be used for the overall running of a business. It can help management define the overall goals they wish to achieve and their overall end products. The program will walk them through on the structuring of a plan to get those products completed. Then, on a daily basis, the managers can work with the computer to maintain management of their firm and projects.

There is also a company workload management program. It gives several different lists; one is a "things to do" list for each person who works for a company. It is arranged according to the priorities of the job. Another list is broken down by each division of a company and arranged according to the highest priorities in the division. Every day a manager can change priorities, delete tasks or add tasks to the system. It gives management an understanding and control over what's going on inside the company. Task sheets can be issued each day by the manager to each individual who does work for him, so they know exactly what the priorities are.

We're planning on developing a simulation language. This would allow management consultants to write a model of a business and its environment, mathematically, describing the environment in terms of probabilities for various alternatives. This simulation can be used to find out the effects of various business decision, instead of actually finding out the hard way. You can use this model to what is likely to happen if you take various paths. An application would be to use various marketing approaches; various items to market, or various publics and you can see what may happen then.

JORJ: Do you rent or sell your software? What limitations are on your software?

JIM: The purchaser buys a license to use the software. If a person would want to rent it, he could do so in the overall lease of the system itself. From our viewpoint a rental would be an outright sale.

JORJ: Once you sell the license for software use, would someone other than the purchaser be able to use the software?

JIM: There are two different types of licenses we sell. One license is to dealers who would receive the right to sell to end users over the counter, but not through the mail. They could sell to all the end users they wanted to that way. The other type of license would be to the end users. They would have the right to use the software only on their own system. Of course, the end user is not restricted from obtaining and using software from other sources.

SYSTEMS

JORJ: How does building a computer system around the AM-100 differ from building a system around another computer?

JIM: The systems analyst has to take a lot of things into account when he picks the type of a computer system a company should have. In general, he has to look at how much a company is going to grow within a five-year period. The system must be big enough to accommodate the five year company expansion. The result is that a person who owns a business is likely to be spending more and getting more of a computer system than he needs at the present time. It's also bad because the prices of computer systems and their various components are going down. So a company would be paying 1978 prices for things it may not need until 1980. The AM-100 is expandable to tremendous proportions. When the systems analyst picks out a system, he can configure the system for the current needs of the company and he can feel confident that as the business expands the computer system can be expanded to fit it, without any changes in the operating procedures that the company has and without any changes in the programming itself. In our first six months in business, most of the work we did was converting programs from one computer system to another computer system. In fact, it was converting from one IBM system to a different IBM system, and it costs a bundle.

JORJ: Could you expand on the AM-100 component flexibility as opposed to say an IMSAI?

JIM: The IMSAI is pretty good. It also has the S-100 bus. One things to look at is the programming language itself. If you want to expand into really big systems with many big disk drives, will the IMSAI handle it? At this exact date the answer is no. Maybe in six months the answer will be yes. For the AM-100 the answer is yes right now.

JORJ: If someone was getting into a small IBM system now, how would that compare as far as expansion?

JIM: Quite often the programs which work on one IBM computer will not work on a different one. There's two types of programming done on IBM computers. One is

the applications program itself which is written, say, in COBOL. Quite often within the COBOL language there's changes to be made, when changing from one IBM system to another, particularly in the input-output statements. That can be a chore. When converting from an IBM-360 to another IBM-360 or IBM-370, you may not have much of a programming change. But converting from a System-32, to a System-3 or from a System-3 to a 360, there'll probably be some conversion changes.

Another type of language is the job control language that IBM has. That allows the computer program to interface correctly with the computer. There's a number of different job control language structures. There's one used on System-3's called OCL. That is completely different from what is used on the IBM 360. The 360's have an operating system called Disk Operating System and there's a set of commands for that. When you expand from that you get into O.S. (Operating System) and there's a set of commands for that. So, if you have several hundred computer programs you've developed over the years in, say, Disk Operating System, and you convert over to Operating System, you have to convert all of this job control language from one to the other, and in many cases it's just an enormous task. Quite often IBM manages to sell data processing managers on upgrading to these different IBM systems because of new advances. Quite often IBM announces they're not going to support the old system anymore. All this has resulted in an expression around the data processing industry "locked into IBM." Now, the AM-100 system allows you to use the same program and the same equivalent to job control language on the tiny system, as you would with one of the huge systems. There'd be very little difference.

JORJ: Can the AM-100 system do everything that a large IBM system can do?

JIM: As far as applications, the AM-100 can handle any type of a business application if it's properly programmed to do so. IBM has been around longer and they have a lot more software for their systems. If someone wants the software tomorrow to do things, IBM could give it to them quicker. I doubt that any business could implement all that software into their operations instantly. Probably by the time a business could assimilate using the software we have, we would have some more software for them that would keep them busy for a while. For most businesses we can do anything an IBM system could do. The AM-100 will keep track of scientific notation and is capable of dealing with large numbers. It has eleven significant digits.

JORJ: Interesting. Basically, where can the systems which you designed be applied?

JIM: In any business. We can supply systems for scientific applications. We have that ability, but our particular plans are geared to the business world. Anything that people need for their own particular business, we can do.

JORJ: Could you elaborate on the word processing you mentioned earlier?

JIM: Okay. One user of word processing is the law firm. Law firms have a tremendous number of legal documents which have to be typed letter perfect. If there's any mis-type, you can't use white out, you have to start over again. Word processing is excellent for this. You type something in, view it on the data entry station, make changes and when it's the way you want, you just press a button and it will print out, perfectly. Another application is on things such as will, where most of the data remains the same and you just change a few items, such as a name and address. You just change the name and address and whatever other items you desire and the system will print out a will without having to retype the whole thing. The price of our system is comparable to an IBM Mag-2 (a word processing machine) and our system does a lot more than that, because you're getting a computer which does word processing and not just a word processing machine.

JORJ: Do you feel there are people who could install a computer and use it to their financial advantage, saving time and money, who are not aware that they could do this?

JIM: Oh, sure.

JORJ: What would the attributes be of someone like that? How big would their company be? What characteristics would these people have?

Another cost factor to look at — the cost of a computer on a 5-year lease is equivalent to an employee working 20 hours a week. . .

JIM: Probably bags under their eyes, for one thing. That would be due to staying up late doing quite a bit of the bookkeeping work. Other typical characteristics would be scraps of paper spread over their living room and divided into piles. Each pile would correspond to an account, such as payroll, for a certain time period. Those would be the main characteristics.

A good criterion to determine if a computer would be economical is, how much per hour would the manager earn if his time were freed from that rum-dum business he has to do and if he were able to put his time into his own business, the thing he's good at. Let's say a guy could net $100 per hour in his own company. If he could save himself four hours of his own time per month he would pay for a computer system. Another cost factor to look at — the cost of a computer on a 5-year lease is about equivalent to half an employee — a part-time person coming 20 hours a week.

Look at the product of a computer, which is information. An example of information would be what magazine would be optimum to advertise in to hit a public between the ages of 18 and 25 who like airplanes, that would yield the most impact per dollar. With information, a manager can make correct business decisions. You can look at the computer as something that would save costs, but you can also look at it as something that would increase profits. Probably the biggest loss to business isn't costs, but is loss of profits that they could have had. These potential profits could be gained by key people in the business having more free time to do the things that earn money for them, and by taking advantage of useful information that would be available.

There's a lot of business people who feel they're earning enough money to meet their needs, but they want to have more time. They want to go home at 5 p.m. and see the wife and kids. Instead, they're staying until 9 or 10 p.m. doing the books, and handling paper work. So, however you look at it, the businessman can increase his output-per-hour by having a computer system which is geared toward his needs.

Data Dynamics Technology
brings to you . . .

"THE CONFERENCES OF MICRO BUSINESS '78"

on cassette tapes

Get just one session, or obtain the entire program.
Learn how microcomputers can be implemented within your business, saving you time and money.

If you attended the session, then order the cassette as a reference. If you missed the session, make an investment in the future of your business by obtaining the pertinent cassette, NOW!

All available programs listed below with price.

"The Form of Small Business Software Packages"
Gordon French, Commodore Business Systems **$16**

"Systems Analysis for Small Business"
Michael Levy, Jethro **$23**

"Practical Low-Cost Software for Microcomputers"
Lon Poole, Osborne & Associates, Inc. **$21**

"Microcomputer-based Business Systems in the Medical and Legal Professions"
Robert C. Brehm, Promedics Data Corporation **$16**

"Some Small Business Computer Users; Their Experience as Customers"
Gene Murrow, Computer Power & Light **$21**

"Looking at the Future of Business Computer Systems"
John Peers, Logical Machine Corporation **$21**

"The Role of Established System Companies in the New Small Business Systems Market"
Matt Peterson, Peterson & Associates, Incorporated **$21**

"The Entry of Calculator Manufacturers into the Small Business Systems Market"
Thomas Bun, CompuMax Associates **$23**

"The Role of Computer Stores in Small Business Systems"
Pete Roberts, Computer Kits **$16**

"Small Business Computer Systems; An Analysis of Products and Markets"
Adam Osborne, Osborne and Associates, Inc. **$21**

"Getting into the Microcomputer Industries"
Robert Jones, INTERFACE AGE Magazine **$14**

"Configuring A Microcomputer System for Business Application"
Bob Harp, Vector Graphic **$7**

"Bringing Your Computer Business On-Line"
Elliott MacLennan, Attorney at Law **$14**

CASSETTE PRICES, as marked
Entire Program (37 cassettes) **$199**

MAIL ORDER INSTRUCTIONS
Circle cassette sessions. California residents add 6% sales tax.
Add $1.50 for the first session and $1.00 for each additional session to a maximum of $4.50 for postage and handling. Foreign shipments add $7.00.
Send Check or Money Order (U.S. Funds drawn on U.S. Bank). Make check payable to: Data Dynamics Technology, 16704 Marquardt Avenue, Cerritos, CA 90701.

Name________________ Address________________

City ________________ State________ Zip________

Total sessions ordered ________________ Amount enclosed $ ________________

DATA DYNAMICS TECHNOLOGY A Division of INTERFACE AGE Magazine

Something else I should mention here is that information has a value, some information has more of a value than other information. An example would be the average number of nostril hairs in a buyer of the services of a company. That would be information without any value at all. Other information of more value would be what caused these buyers to buy. Part of a good system analyst's job would include the determination of what kinds of information would be most valuable to a business.

JORJ: Exactly what do you do and what does your company do?

JIM: I run the overall business. In terms of the client, I work with the client to help him determine his exact needs right now, and what he may need in the future. Then we tailor a system so he can meet those needs.

JORJ: Do you also tailor the software to fit each individual client?

JIM: Sure. Let me summarize this, as far as the overall hardware choice objectives. There should be something the company can start with, and something that would allow for growth as the company expands. Ideally, one would get an optimum return for the dollar. The AM-100 is geared toward starting small and expanding.

The software allows the computer to service the needs of the company itself. The computer is just a machine. The software should align with the objectives of the company itself, and take the drudgery out of work so that creative human beings can do less tedious tasks and do live communication functions which only a person can do.

The AM-100 fits into these software objectives with its programming ease. The AM-100 is just faster to program. You can get a lot more programming done per dollar. Programming speed goes up and cost of software goes down. This would be most noticed with customized software, where you're paying a programmer by the hour.

For a business that is intending to expand beyond a minimal size of 100-200 transactions a month, the computer results per dollar invested, or return on investment, is greater with this system than any other thing I know of. For really small businesses this may not be the case. Up to really good sized businesses, the AM-100 is about the best thing going.

One thing I want to throw in here is that the computer is really a tool. The results you get from a tool have to do a lot with the skill of the user of the tool. An important function to keep in mind is how good of a systems analyst's job is done. How well the hardware and software are configured to align with the goals and existing scene of the company. That would include how good the training program is. Is the computer a complex tool to learn, or is it really simple? I'm dealing with another computer other than Am-100 now, and it is extremely complex. Ours is extremely simple.

JORJ: Thank you very much, Jim.

JIM: You're welcome.□

ABOUT THE AUTHOR

James W. Kitzmiller is the president of Kitzmiller Systems, Los Angeles, California. He received his B.S. degree in Electrical Engineering from the Case Institute of Technology; his M.S. from Ohio State University, and his Ph.D. in Management Science from Arizona State University.

Jim has been working with computers since 1965. He has worked as a salesman, programmer, systems analyst, business manager, and has worked in the field of human relations.

GLOSSARY OF TERMS

AM-100: A computer made by Alpha Microsystems in Irvine, California.

BASIC: A well-known computer language.

BIT: A 0 or 1.

BOARDS: Physical devices that plug into the computer and perform various functions, such as input/output, memory, and actual processsing.

CHARACTERS: Symbols, e.g., letters of the alphabet, numbers, @, &, and so on.

COBOL: Acronym for COmmon Business Oriented Language, a computer language.

DISKS, HARD DISKS, BIG DISKS: Physical memory storage devices. Disks are flat surfaces, like 45 RPM records.

DISK DRIVE: A device which allows the computer access to information which is on disks.

8080: A type of central processor.

HARDWARE: The physical part of a computer.

IBM SYSTEM-32, 360, 370: Types of IBM computers. They are listed here in order of decreasing size.

IN-HOUSE: Contained completely within the business. An in-house computer operates without having to connect to another computer over phone lines.

INTERFACE: A go-between. Something that allows one computer component to talk to another.

MICRO, MICROCOMPUTER: A computer that uses 8 bits as the length of words for its instructions.

MINI, MINICOMPUTER: A computer that uses 16 bits as the length of words for its instructions.

ON-LINE: Connected to the computer. Often, connected to the computer by a telephone "line."

R.P.G.: Initials standing for Report Program Generator, a computer language.

SOFTWARE: The instructions that tell the computer hardware what to do. Examples are payroll and general ledger programs.

SYSTEM: People, machines, materials, and procedures and/or policies arranged to produce a desired result.

SYSTEMS ANALYST: A person who studies systems in depth and comes up with improvements.

TERMINAL: A device that sends and receives information to and from the computer.

WORD PROCESSING: Automatic typing. It is very useful for form letters, and retyping documents when there are only minor changes in them.

Consideration for Computer Implementation in A Small Business
Part I of 4 Parts

By Roger Williams

INTRODUCTION

The usefulness of a computer in a small business environment is self-evident. For any business with a yearly gross ranging from ¼ to 5 million dollars, the primary question is not *whether* to implement a computer, but rather *when* and *how*. The small businessman frequently is unfamiliar with, and sometimes intimidated by, the complexities and mystery associated with computers. The purpose of this article is to penetrate this mystique so that the businessman can evaluate objectively when the computer is economical for his own needs, and then how to proceed.

The first two parts of this article are dedicated to the question of *when* to implement a computer, considered from the perspective of both general and detailed economics. PART 1 discusses the general economics of computer implementation. It first presents a rudimentary concept of differential cash flow analysis, which it then uses to describe the most general effects of computerizing. It then proceeds with the analysis by describing the economic benefits and the costs of computer implementation, including both hardware costs (costs of the computer itself) and software costs (costs of the programs). PART 2 discusses the detailed economics of computer implementation. It refines the concept of differential cash flow, then numerically analyzes a hypothetical business in detail, complete with graphs and tables. It also presents guidelines to enable the small businessman to apply the same analysis to his own business.

The last two parts are oriented to the question of *how* to install a computer once the decision is made to do so. PART 3 discusses the problem of selecting and optimizing the computer system in terms of both hardware and software. PART 4 describes what happens after the computer is plugged in. It discusses the capabilities of the computer and describes how the bookkeepers and managers interact with the programs. It also suggests safeguards and pacing to ensure a smooth transition from manual operation to computer operation.

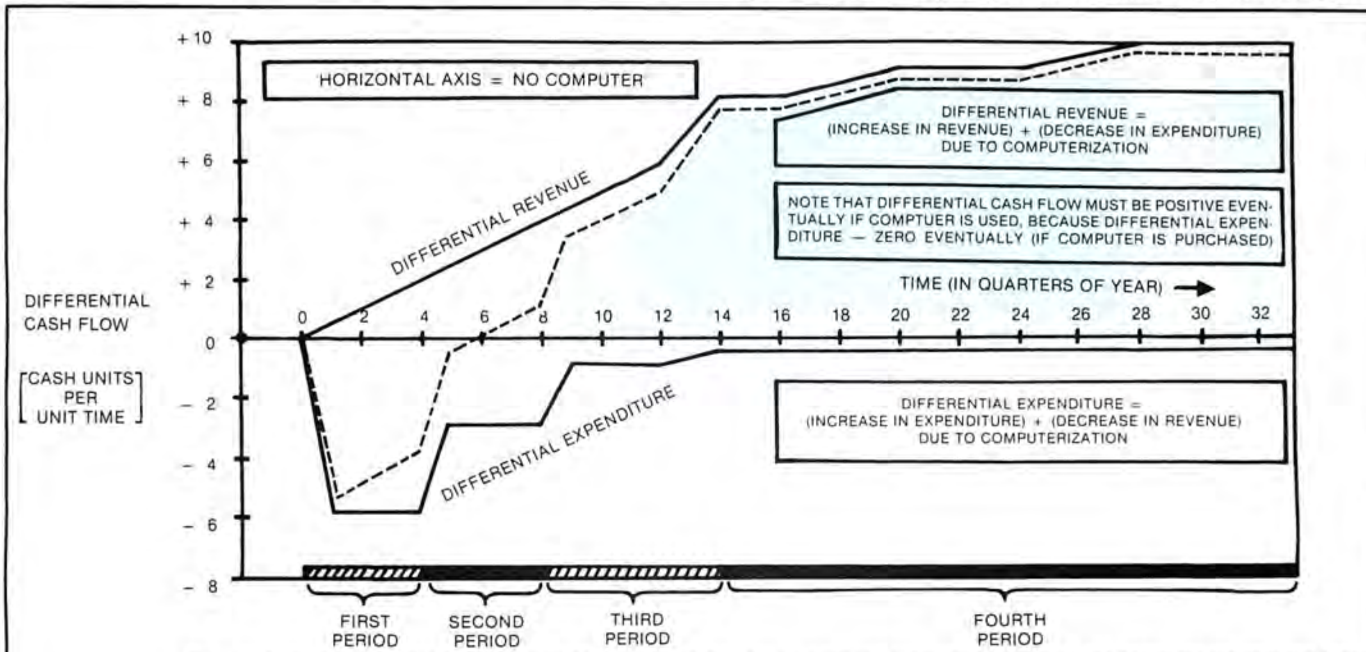
PART 1 — THE GENERAL ECONOMICS OF COMPUTER IMPLEMENTATION

The general discussion of the economics of computer implementation first describes the differential cash flow method of analysis, then utilizes this analysis, in very general terms, to describe the expected results of a computer implementation for most businesses. It then shows that the computer is almost certainly *viable*, and that *affordability* thus becomes the primary determinant of when to computerize. It also shows that computer implementation is rendered most affordable by the proper balance of cost versus capability, which is fortunately not difficult, as disclosed by the fact that hardware and software costs are both well-defined and stable within a narrow dollar range.

Differential Analysis and the Expected Results of Computer Implementation

The analysis used in this article is *differential analysis*, in which the economic effects of implementing a computer are described by analyzing differences in cash flow resulting from the implementation in comparison to no implementation. This analysis first requires certain definitions. Define *differential revenue* as the increase in revenue or the decrease of expenditure generated solely by the benefits of the computer, such as savings of labor costs. Likewise, define *differential expenditure* as the increase in expenditure or decrease in revenue generated solely by the implementation, such as the cost of the computer itself. The difference between these two quantities is defined as *differential cash flow*, which indicates the effects of the computer implementation in a concise manner. These three definitions are illustrated in Figure 1.

Next define *differential accumulated cash* as the cash accumulated over a specified time interval as a direct result of the differential cash flow — or equivalently, as a direct result of the computer implementation compared to no implementation. This differential accumulated cash is calculated by multiplying the differential cash flow times the time-interval, just as one would calculate the accumulated water in a bucket by multiplying the water flow (rate) times the time-interval. If the dif-



This graph shows differential revenue, expenditure, and cash flow for a typical computer implementation. The scales are arbitrary and exaggerated for the sake of illustration. The computer is implemented at time = 0, and is gradually utilized for increasing work loads as costs decrease through four time periods discussed in text. The horizontal axis is the no-computer reference for differential quantities.

Figure 1.

ferential cash flow is not constant, the calculation is slightly more complex, but the concept remains intact. Figure 2 illustrates the results of the calculation applied to the differential cash flow of Figure 1.

Using these definitions, one can describe in quite general terms the expected economic effects of any computer implementation, independent of the specifics of the business. For example, as illustrated in Figure 1, the differential cash flow will be somewhat negative for a first time period (perhaps a year) because the computer is costing dollars but is not doing very much work. Then during a second period (perhaps the second year) the differential cash flow starts climbing rapidly to become positive as the computer is phased into regular operation and as costs decrease. During a third period (perhaps the third year) the differential cash flow continues to climb rapidly as costs drop nearly to zero, and as the computer does increasing amounts of work. Finally, in a fourth period (perhaps the fourth year and beyond) the differential cash flow continues to rise, but much more slowly, as the situation stabilizes and benefits increase slowly with the growth of the business.

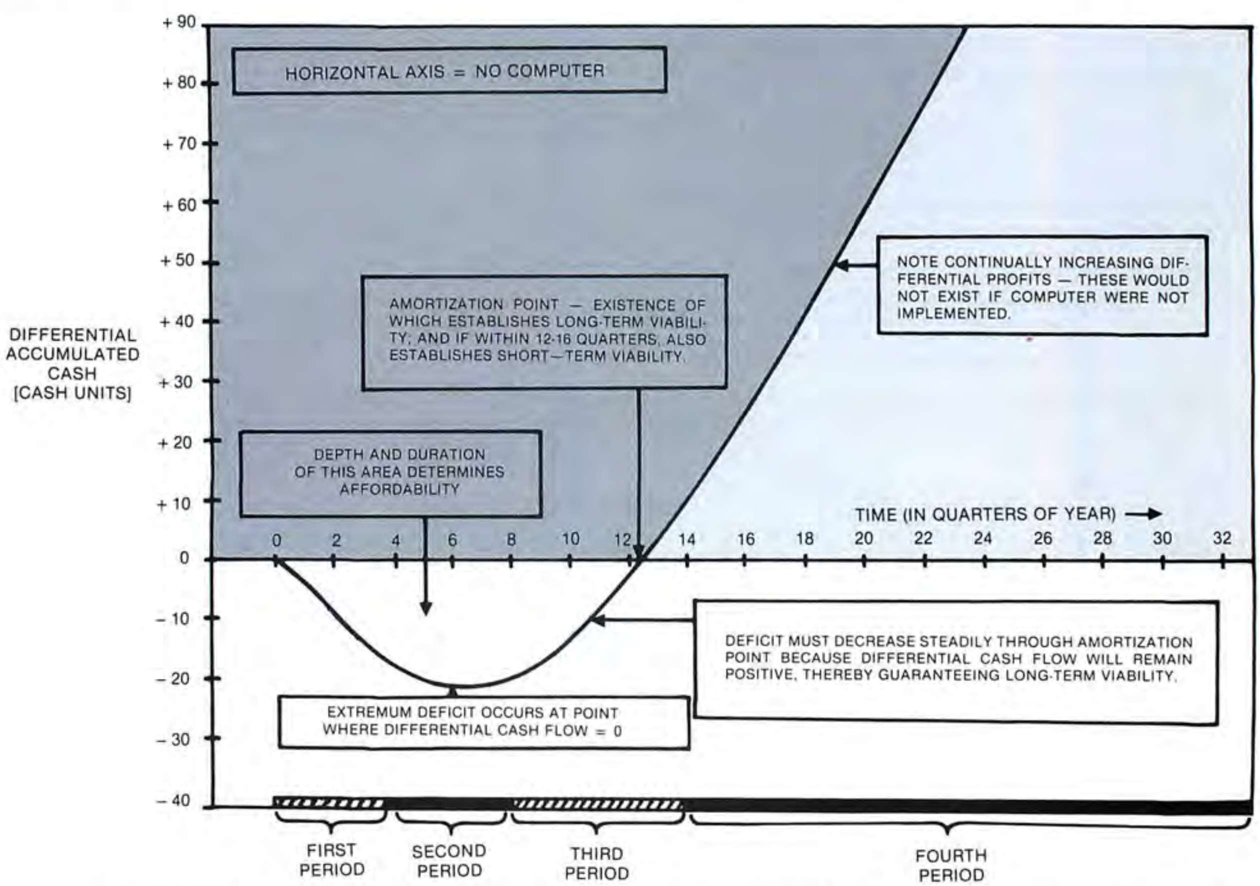
These same general features can be evaluated in terms of differential accumulated cash, as illustrated in Figure 2. A cash deficit develops rapidly over the first time period, then this deficit reaches an extremum followed by a decrease during the second period. This deficit continues to decrease until it reaches zero during the second or third time period — the point at which the investment is amortized. Subsequently, a cash surplus develops rapidly and continues to do so at a rate slowly increasing with time.

This differential accumulated cash analysis is extremely powerful, for *it reveals that profits, in real* dollars, *must* result from implementing a computer. To see why, note that the differential cash flow ultimately will become positive if the computer is used at all, because monthly costs eventually will drop nearly to zero. This means that the differential accumulated cash eventually must reach the zero, or amortization, point (albeit after a possibly long time). Thus the computer is almost certainly a *long-term viable* investment, in the sense that it will amortize itself at some time. If amortization can be reached within three to four years, the computer may be regarded as a *short-term viable* investment.

The power of the analysis is even further emphasized by considering the consequences of merely delaying an implementation of the computer, even if the investment is viable only in the long term but not the short term. As illustrated in Figure 3, such delay would also delay the differential accumulated cash, creating a *perpetual* difference in accumulated cash as evaluated at *any* time after the first few years. These irreversible costs of delay are clearly illustrated in Figure 4, which compares the accumulated cash flows of different delays in comparison to that of no delay. Note that there exists, in the long term, an eternal and constantly increasing deficit of differential accumulated cash.

This is real cash, not just paper! It is with this perspective in mind that a decision *not* to implement a computer *immediately* must be considered most carefully as a high-risk option containing a high probability of irreversible and ever-increasing differential losses.

Of course, the possibility exists that a small business



This graph shows differential accumulated cash for the typical computer implementation of Figure 1. The calculation is described in the text, and is known technically as a "time integral" of the differential cash flow. The scales shown are arbitrary and exaggerated for the sake of illustration. The concepts of viability and affordability are discussed in the text.

Figure 2.

cannot *afford* a computer, no matter what the consequences may be in terms of differential losses, and therefore must accept the losses. This situation might occur if the cash resources of the business could not accommodate, without undue strain, the initial negative differential cash flow resulting from the costs of the computer. This is a classic situation arising from under-capitalization — where a potential for large profit exists but cannot be utilized for lack of capital.

Since it is clear that the computer would be long-term viable, and probably would be short-term viable, the most immediate question is when a computer implementation would be affordable. This requires a more detailed analysis of the expected benefits (differential revenue) of the computer in comparison to the costs of implementation (differential expenditure). In terms of differential analysis, the differential cash flow must be determined, and then must be evaluated for when the business resources could accommodate the initial negative differential cash flow periods. It is worth noting that an affordable implementation is necessarily short-term viable (unless the business is over-capitalized enough to support long-term negative cash flows!).

In other words, the rational decision to implement a computer is already clear merely by establishing affordability, and nothing else. The question is not whether to implement a computer (the answer is already "yes"), but when — the answer being "as soon as the computer is affordable."

The most favorable condition for affordability is that the extremum accumulated cash deficit be as small as possible. This requires that the benefits (differential revenue) be as high and as rapid as possible, and that the costs (differential expenditure) be as small and as short in duration as possible. These factors obviously conflict — a low monthly cost of the computer results in a low capability of the computer to produce economic benefits, and vice versa. An optimum balance between cost and performance thus exists, and the objective is to achieve this optimum by a proper selection of components, as will be discussed in detail in PART 3.

Independent of details, however, a specification of this balance is not as difficult as one might expect, and establishes a well-defined and narrowly-bracketed cost of the computer implementation. A proper consideration of hardware and software will determine a $7,000 to $10,000 end cost for hardware, a $2,000 cost for software, and a $600 per month cost for custom programming for a year. These amounts are, surprisingly, somewhat independent of the size of the small business, as discussed below.

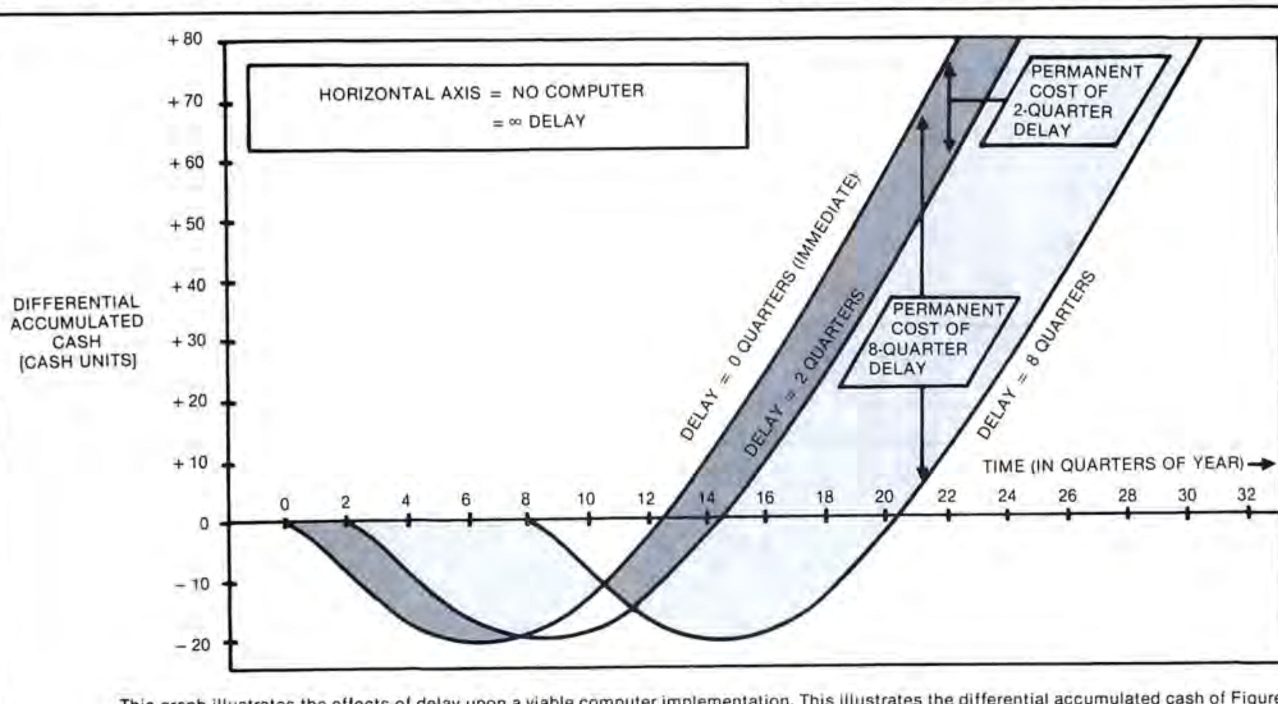
Once having determined the approximate cost of implementation (differential expenditure), the benefits of implementation (differential revenue) must be evaluated. Although these are not as well-bracketed as the costs, some estimates may be given. Labor savings will range from ½ to 5 employees, representing $300 to $3,000 monthly. Increased sales and improved management will account for 2% to 4% additional gross profits. Again, details are given below.

These amounts of differential expenditure and revenue, if estimated for the specific business, can be used to calculate the differential cash flow and differential accumulated cash generated by the computer implementation. This in turn, as outlined above, will determine both the viability and affordability of the computer. Such a procedure is utilized in PART 2 to analyze a hypothetical business to determine when the computer is affordable and amortizable within 3 to 4 years, and to elucidate what cash results occur, and what would be the permanent costs of delay.

The remainder of this PART 1 will be dedicated to clarifying the above statements of expected differential expenditure (computer costs) and expected differential revenue (computer benefits), especially in terms fo their relatively well-defined nature independent of the small business.

Differential Expenditure — The Costs of A Computer Implementation

The costs of implementing a computer depend, first of all, upon whether the business premises are to con-



This graph illustrates the effects of delay upon a viable computer implementation. This illustrates the differential accumulated cash of Figure 2 delayed 2 quarters and 8 quarters. The effects are irreversible and steadily increasing. The horizontal axis is the no-computer (∞ - delay) reference for differential quantities.

Figure 3.

tain a complete computer system, or instead are to contain only a terminal which is connected via a telephone line to a remote "time-sharing" system.

The Time-Sharing Option

The time-sharing option, although popular in the past, is rapidly becoming obsolete for the small business. The reasons become apparent upon an examination of four factors, including the cost of the terminal, the cost of connection to the remote computer, the cost of mass storage of business data, and the cost of software.

The end cost of a terminal with CRT (a TV-type display console with a keyboard) and printing capability (a necessity for a small business) is about ⅓ the cost of a complete computer system. The monthly rental would range from $125 to $175 per month. Added to this must be the cost of connection to the remote computer. The telephone line would cost about $1 per hour and the "connect" time would cover over $5 per hour during business hours. These hourly costs, assuming a 4-hour day connection, add up to $24 per day, or to nearly $500 per month.

The cost of mass storage is based upon the amount of data stored at the remote site. It is important to note that data cannot be transmitted quickly from the terminal to the computer or vice verse, nor is there mass storage capability at the CRT terminal accounted above. Therefore, the remote computer must accumulate all data through at least an accounting quarter, with an archival printout at the end of each quarter. (Note that this renders all future computer access to this data impossible!). The amount of data thus accumulated easily can fill the equivalent of "two floppy disks." Such storage, about 500,000 "bytes," would cost about $150 to $250 per month.

Next, software costs must be considered. Although a time-sharing facility may save the initial cost of certain programs, these programs cannot be modified for custom use, and additionally all custom programming for special purposes must still be bought. The initial savings of $1,000 to $2,000 in such software is a marginal savings over the in-house computer, being compensated within a few months.

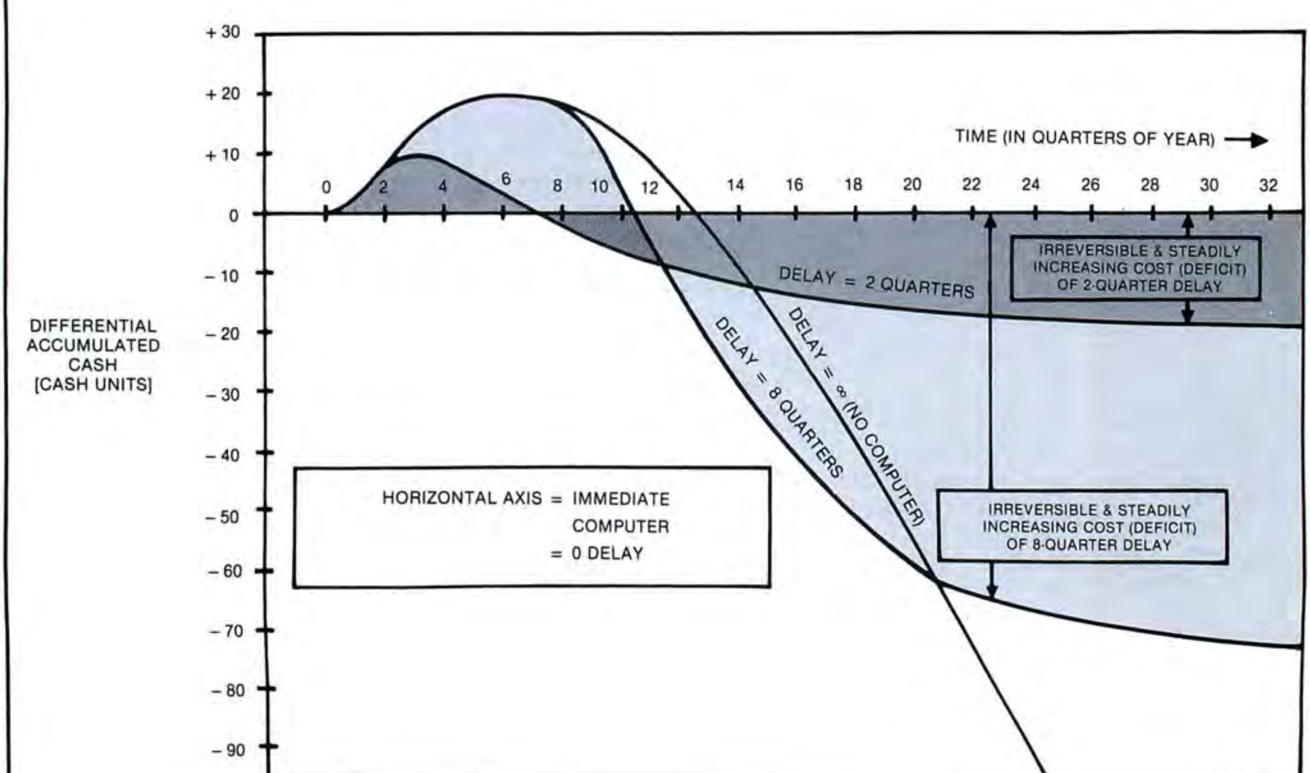
In evaluation, the hardware costs of the time-sharing alternative total $800 to $1,000 per month on the assumption of 4 hours per day usage. If usage drops even to ½ hour per day, the costs still remain at $400 to $500 per month, because the data must still be stored and the terminal must remain at the business premises. On the other hand, if usage climbs to 8 hours per day — such as might be needed to accommodate custom program development — the monthly cost could well soar to $1,200 to $1,400 per month.

As already stated, and as detailed below, an in-house computer hardware might cost a maximum of $10,000 or perhaps $500 per month, depending upon financing. The time-sharing option would then cost, on a monthly basis, at least as much as the in-house computer, even if hardly used, and easily could cost twice as much. In addition, these costs are irretrievably lost rather than converted to an asset, as would be the case in a purchased computer.

In-House Computer Implementation Costs

Now the path is clear on the question of time-sharing vs. the in-house computer, at least for small business applications with access to normal commercial time-sharing facilities, and on the assumption of the above specified end costs for the in-house computer system.

These end costs are now examined in greater detail by



This graph is identical to Figure 3 except that the 0-delay (immediate computer) option is used as the reference for differential quantities to define the horizontal axis. This change of perspective clearly emphasizes the effects of delay as an irreversible and steady increasing deficit of differential accumulated cash.

Figure 4.

analysis of general hardware costs, general software costs, and the stability of these costs for the next few years.

Hardware Costs of the In-House Computer

Practically all small businesses will require the same general hardware, a paradoxical situation created by the fact that the minimum set of *essential* hardware is also grossly under-utilized in the smaller businesses. (This also means that the computer will accommodate growing needs without additional investment!) A minimal hardware configuration in any business consists of the mainframe, or computer itself, plus a printer, a CRT terminal, and a mass storage facility.

More specifically, the mainframe needs to have at least 40 "kilobytes" of memory to accommodate the necessary programs. The printer should have a "pin-feed" capability for correct printing on pre-printed forms. The CRT terminal must have at least a highly readable upper and lower case display of 24 lines by 80 characters per line. It should have a numerical keyboard in addition to the standard typewriter-style keyboard. The mass storage should consist of least a dual floppy disk drive with a capacity exceeding ½ million "bytes" — to facilitate copying disks and to provide at least a minimum storage needed for any business larger than a one-man operation.

The cost of the mainframe ranges from $2,000 to $2,500 including memory, input/output boards, and incidentals. The printer ranges from $1,000 to $1,500 including the pin-feed platen. The CRT terminal costs $1,000 to $1,500. The dual floppy disk drive with a half-million byte storage, including the "controller" and incidentals costs $2,000 to $2,500. Thus the cost for a *minimum* suitable hardware system is from $6,000 to $8,000, averaging $7,000. These costs are quite independent of specific brands and features.

The cost for a *maximum* system, of course, can climb without limit. However, a $10,000 cost would upgrade the capabilities beyond any reasonable needs of a small business. Already most components are under-utilized. For example, the printer could print a book every day if desired. The CRT, if used all day, would accept huge amounts of data. The mainframe needs no more memory or other capabilities, for it already can calculate 10 to 100 times more than is needed in a small business. Data manipulation and storage are the primary functions needed. The only limitation is with the mass storage capability of the disk drive, and even this can be circumvented if the user is willing to exchange disks frequently. (The disk can be removed or inserted into the drive as desired, just as a tape cassette might be.) If desired, a disk storage of 10 to 100 times the above capacity could be purchased within an extra $3,000 initial cost.

If a multi-terminal system is needed, then the extra CRT and necessary hardware and software could be purchased within a $3,000 extra initial cost. Thus one can say safely that a $7,000 to $10,000 cost of hardware will cover the needs of all but the largest of the "small businesses," and that even these may be accommodated with minor inconveniences such as frequent disk exchanges.

Software Costs of the In-House Computer

The software costs are next discussed. There are two basic types of software needed. One type, called operating software, functions to coordinate the CRT, printer, and floppy disk drive to the computer itself, and is often supplied with the hardware. The second type, called applications software, performs the actual work the user requires, such as accounting. Suitable applications packages for the small business will cost $1,000 to $2,000 for general ledger, payroll, accounts receivable and payable, and inventory-control capabilities. Although such applications software may not do exactly what is wanted, and is seemingly expensive, it is nevertheless highly cost-effective. It could be a costly mistake for the small business to attempt large-scale custom programming for these tasks.

The direct cost of custom software development is often underestimated. The industry-acknowledged development cost of programs, after all documentation, debugging, and implementation are completed, exceeds $10 per line of code. Thus custom applications programs could cost well upwards of $10,000 or more.

An indirect cost of custom software is the long time delays before implementation in the computer, thus causing permanent effects upon the accumulated differential cash in the same manner as discussed for delaying computer implementation. These delays can be far larger than most nonprofessionals realize, upwards of several months sometimes.

It is thus clear that as much of the desired software capability as possible should be purchased as applications packages. Custom programming, although essential for implementation of such packages, should be restricted to such implementation, and to programming applications for which packages are not available. Additionally, such custom programming is best done gradually. A half-time programmer working for a year, costing about $600 per month, is both necessary and sufficient to accomplish the most-needed custom work and adjustments, including even a few complete major programs. This pacing also gives the business sufficient time to adjust to the computer and define what it needs from the programmer. This helps to avoid premature specifications which are programmed then become quickly obsolete, potentially wasting substantial sums of money. Also, programmers may well produce ⅔ of their work during the first ½ of each day, as programming often benefits from "overnight solutions" to problems.

Cost Stability for the In-House Computer

One last observation considers the stability of hardware and software costs during the next few years — that these costs are likely to be similar a year or two from now. The small business will not save substantially by waiting a year.

It is sometimes thought, based upon the observation of the plummeting costs of pocket calculators and hobbyist computers over the last few years, that a complete business system will plummet likewise — that within a year the same capability may be purchased at half the cost. The fallacy of this thinking is clarified by examining the cost stability of both software and hardware.

The software costs of a computer system, as discussed, include those of both applications programs and custom programming. Although many low-cost applications programs may proliferate, these are usually of limited capability. High-capability software — essential to creating high differential revenue — is decreasing in price only very slowly. However, even if such were to halve in price overnight, the savings would represent only 5% to 10% of the system cost. Furthermore, custom software is increasing in price fairly rapidly, due to inflation and to the exploding demand and decreasing supply of professional programmers.

The hardware costs of the computer system include the cost of peripherals such as the CRT, printer, and floppy disk driver, and the cost of the mainframe. It is worth noting that over the last three years, nearly every peripheral and mainframe either has remained the same price or has increased. Only memory boards and the CPU

have decreased significantly. These represent such a small portion of the total system, that even halving the price creates only a 3% savings on the total business system. The peripherals and mainframe utilize mechanical construction, moving parts, and large amounts of conventional electronics, all of which are subject to inflation and will not benefit substantially in the next year or two from the type of micro-electronics used in handheld calculators and in memory boards or CPU.

Much talk has been centered on the new technology of magnetic bubbles and CCD units. Again, these are not relevant to reducing small business system costs. If these are used for memory, only a minimal cost effect is realized, as already noted. Neither technology is going to reduce the cost of mass storage significantly for the small business. Both are far more expensive per "byte" than the floppy disk complete with driver, and will remain so for several years. Even if then cost competitive, they will not soon replace the disk. The CCD is incapable of providing a permanent storage without constant power-on, and neither the CCD nor the magnetic bubble is capable of replacement and filing for archival storage as is the floppy disk (which may be stored like a phonograph record). In other words, neither is capable of providing a long-term storage at a cost of $30 per million "bytes" like the floppy disk, separate from its driver mechanism.

It is clear, then, that the cost of implementing a complete business computer system will remain stable for a few years — and that the possibility of rapidly decreasing costs is an unrealistic hope which is extraneous to the analysis of *when* to implement a computer. Perhaps even more important, even if computer implementation costs were to halve in the next few months, the cost of such delay could be five to ten times higher than any savings, and these costs would be permanent and irreversible, as emphasized in the previous discussion of differential cash flow analysis.

Now that the cost of implementing a computer system (differential expenditure) has been clarified, the benefits of implementation (differential revenue) remain to be elucidated.

Differential Revenue — The Economic Benefits of A Computer Implementation

The most obvious benefit of installing a computer is the reduction of labor costs relating to invoicing, billing, payroll, general ledger, accounting, and inventory control. The computer would eliminate redundant entries in a multiplicity of journals and would eliminate tedious calculations and error checking.

A less obvious savings would accrue through improved management enabled by the computer. Critical situations may remain undetected with the quarterly or yearly evaluations available with manual accounting, but the computer provides instantly accessible business data and calculated parameters. Inventory costs especially would benefit from such improved management by increasing turns ratios. Most likely a 5% to 10% better utilization of invested capital would result, creating a 2% to 4% improvement of gross profits.

In addition to analysis of current transactions, the computer can generate future projections of accounts payable and receivable, cost of capital, cash flow, budgeting, inventory control, and critical business parameters. An especially good example would be the type of differential analysis utilized in this exposition as applied to any business option, not only computer implementation.

Another service especially valuable to complex business operations is that of critical-path analysis to highly interdependent processes such as manufacturing and contracting. Without such a tool, a business might be vulnerable to critical processes not being completed and thereby causing severe delivery and cash flow problems.

Gross receipts may be directly *increased* with the use of the computer to *improve* sales management. Advertising campaigns could be analyzed for effectiveness. Motivational programs for sales personnel may be implemented in a manner impossible without a computer, especially regarding performance analysis and instant feedback to the personnel. Clients could be given personalized attention using data recorded and analyzed with the computer, as with periodic contacts. Certainly a 1% to 5% increase in gross receipts could be realized, with a similar increase in gross profits.

An intangible, but nevertheless real, savings can be achieved by the computer used for sorting and prioritizing task lists so that important tasks are not delayed or ignored for the sake of less important ones. Such might include client contacts, errands, deadlines, and messages — all of which would be worth money to organize properly. This alone might reduce labor requirements as non-essential tasks are rationally procrastinated, then discarded.

A huge potential savings — perhaps even the whole business itself — results from the ease with which the computer can make copies of vital business records for safekeeping, both in the form of magnetic disks and printout. These records are often never duplicated in a small business, thereby are vulnerable to destruction through fire, theft, or other disaster. If done, manual duplication of these records is quite costly, a situation creating another direct savings from the computer.

Finally, a seldom contemplated source of income from a computer installed in an existing small business is providing computer services for associates for a nominal fee. For example, a manufacturer or importer often has a complete network of sales representatives, each functioning as an independent business. Local associates might desire to utilize an already existing computer system for their own needs, especially in the context of a well established working relationship. Also, neighbors, such as members of a local merchants association, might rent computer facilities, especially if they are unable or reluctant to invest in their own computer.

This extension of the computer to outside users, if done on a small scale, merely effectively utilizes otherwise idle time and does not require any additional investment. If more such revenue is desired, the computer could be upgraded to a multi-user, multi-tasking capability on the premises with very modest extra cost, as 90% of what is needed is already implemented.

These benefits of computerizing represent differential revenue which, when combined with the preceding differential expenditure, form the basis for a rudimentary differential cash flow analysis which can be utilized to evaluate both the viability and the affordability of computerizing for any small business.

SUMMARY

In summary, the question of when to computerize becomes a most important question, which if ignored, can create most costly and permanent deficits of cash in comparison to what could be — to the potential of the business. The costs of implementing a computer are reasonably well bracketed to the same amounts for all small businesses. The benefits described are potentially extensive, and must be evaluated by the individual small businessman. The tool of differential cash flow analysis can be utilized quickly to provide a feeling for the results of computerizing, both in terms of immediate affordability and long-term cash results.

PART 2 utilizes these concepts to examine a hypothetical business in detail, refining the differential analysis to include cost of money, return on investment, depreciation, inflation, and tax factors.☐

MICRO WORLD MAY SPECIALS

Integral Data Systems Printer

$749
reg. $799

low cost, professional performance

Meet the IP-125. Ideal where you need a space-saving hard copy printer for your office. Microprocessor controlled, uses 8½" roll or fan-fold paper. Instantaneous print rate to 100 cps, sustained speed of 50 cps. Just 24 lbs.

- Upper, lower case
- RS232 serial interface
- 7 x 7 dot matrix
- 256 char. multiple line buffer
- Tractor feed option $150

Low Introductory Price! **(add $10 shpg.)**

Hazeltine 1500

$995
reg. $1125

Kit

(add $20 shpg.)

value packed—every feature is standard!

The high-quality basic terminal with extra standard features you can't find on most CRTs. Plus the features you expect, like 24 x 80 character display, auxiliary serial port, 10-key numeric pad. At this low price, you can order two.

- Standard & reverse video
- Upper, lower case
- Cursor addressing & sensing
- 7 x 10 dot matrix

Low Introductory Price!

Cromemco Z2-D single disk drive

fast, commercial-grade computer

$1345
reg. $1495

(add $25 shpg.)
Kit

Take a step forward with this thoroughly professional unit designed for engineering, science, production, business, education. To make it even more enticing, you save $150 if you buy now.

- Fortran & Basic available
- 21-slot motherboard, 30-amp power
- Z-80 processor
- each 5" disk stores up to 92K bytes
- shown with optional aluminum cabinet, $195

NEC Spinwriter

$2995
reg. $3235

(add $25 shpg.)

Low Introductory Price!

20% faster, 50% longer life, 10% quieter

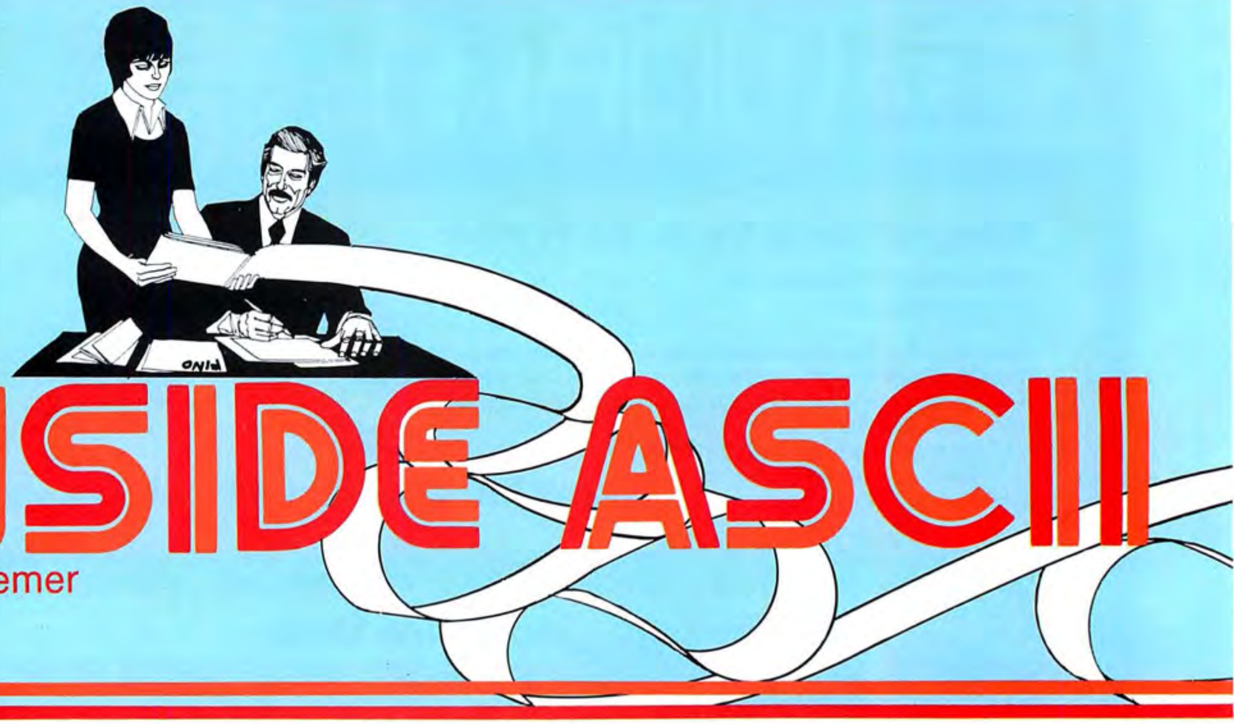
...than Diablo-type impact printers. Microprocessor-controlled, tractor feed, camera-ready print quality, forms control, proportional spacing, bidirectional printing, operator-interchangeable standard/foreign language type fonts.

- 50 million characters between failure
- 55 cps print speed, RS232 interface
- graphics with 5760 points per square inch
- rapid snap-in ribbon cartridge

MicroWorld (formerly Byte Shop Mail Order) is a division of The Phoenix Group, Inc., which operates five retail computer stores and the nation's largest computer wholesale business. MicroWorld has the resources to serve you. Call or write for all your computer needs.

MICRO WORLD
803 N. Scottsdale Rd.
Tempe, AZ 85281
(602) 894-1193

New! Byte Shopper Guide Vol. III, $3.95 (includes $1.00 postage)



INSIDE ASCII

By R. W. Bemer

The data alphabet called ASCII (Figure 1, page 98, and Reference 1), also has two other names—International Standard 646 (the ISO Code [Reference 2]) and Alphabet No. 5 of CCITT (the International Consultative Committee for Telephone and Telegraph). It is used throughout the world, incorporated in billions of dollars of equipment.

But is it used correctly and wisely? Not always. There are misinterpretations, and gaps in definition that permit nonstandard usage. This article (in three parts) will give you the background, peculiarities, preferred practices, and new developments for ASCII. You will find a lot of information not too generally known or realized; it should help in the correct and safe usage of ASCII. For additional help, you can reference the various national and international standards given in Table 1. Some other detailed articles are listed in References 3, 4 and 5.

	ISO	ECMA	ANSI	FIPS PUB	CSA	BS	AS	CCITT	JIS	GOST
Binary-coded Character Set	646	6	X3.4-1977 $4.50	1	Z243.4	4730	1776	V.3	C6220	13052-67
Graphics for Control Characters	2047	17	X3.32-1973 $3.50	36		4730				
Character Set for Handprinting	97/3 N119		X3.45-1974 $5.75	33	Z243.34.1					
Additional Controls Character Imaging		48	BSR X3.64							
4-bit Sets	963	14		15	Z243.6	4731/1	1070			
Code Extension Techniques	2022	35	X3.41-1974 $6.00	35	Z243.35	4953				
Registration Procedures for Escape Sequences	2375									
8-bit Coded Character Set	DIS 4873	43	X3L2/77/08							
Character Set for 7 x 9 Matrix Printers		42								
Keyboard	2530	23	X4.14-1971 $3.75			4822/1	1922			
Character Sets for Programming Languages	97/5 N436	53								

Legend

ISO - International Standards Organization
ECMA - European Computer Manufacturers Association
ANSI - American National Standards Institute
FIPS - Federal Information Processing Standard
CSA - Canadian Standards Association
BS - British Standard
AS - Australian Standard
CCITT - Consultative Committee International, Telephone & Telegraph
JIS - Japanese Industrial Standard
GOST - USSR Standard

Table 1.

STICKS 4-7

ASCII, as a 7-bit code, is usually represented in 8 columns of 16 positions. The row positions are 0000 through 1111, the low-order 4 bits, 0 through 15 in decimal. The columns are 000 through 111, the next higher 3 bits, 0 through 7 in decimal. For some reason, the developers of ASCII found it convenient to refer to these eight columns as "sticks." So shall we. Each position will be represented in this article by its usual decimal representation. For example, capital A is position 4/1. Figure 2 is a representation of ASCII that is more convenient to those working in octal, rather than hexadecimal, notation.

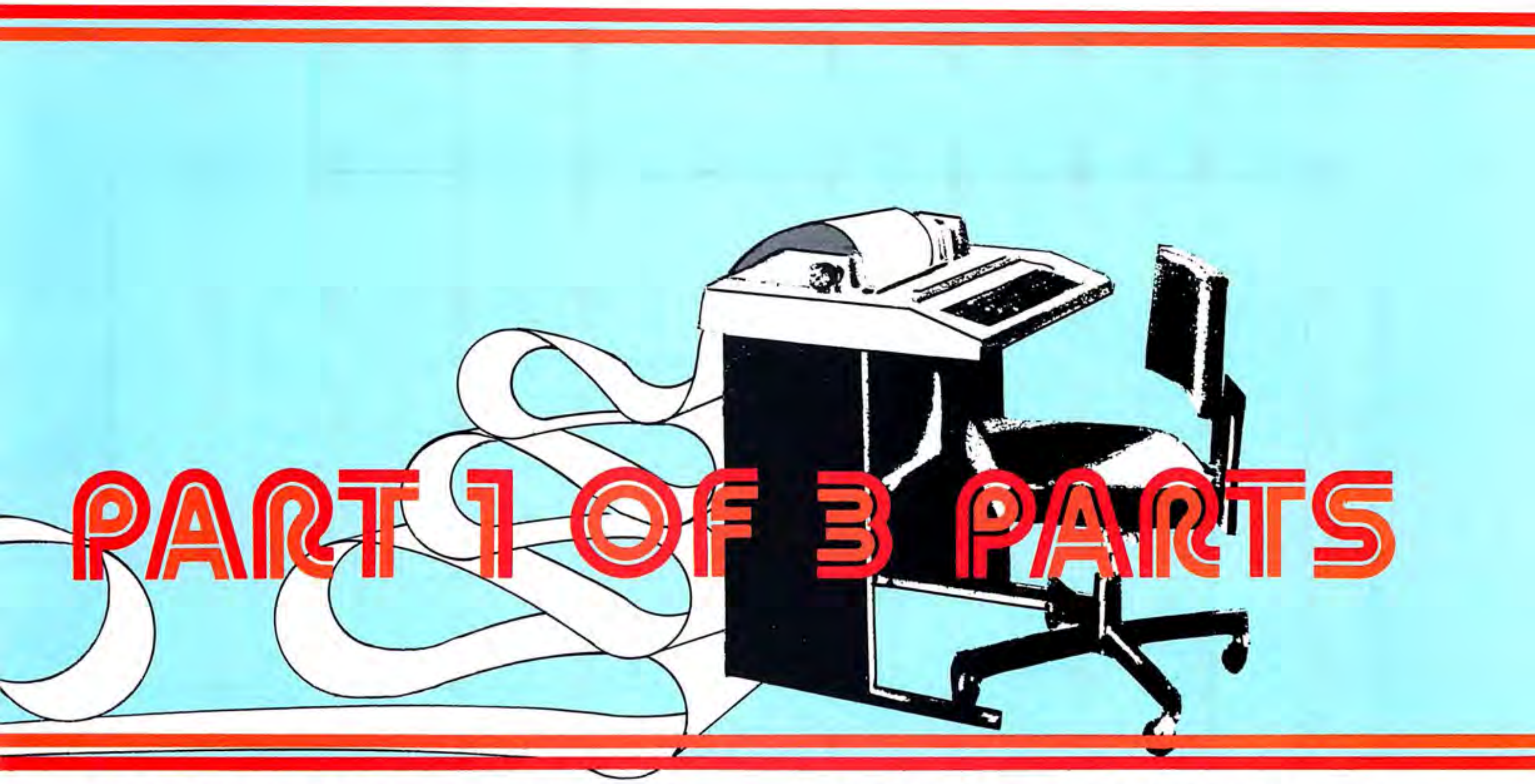
HIGH ORDER OCTAL DIGITS → / LOW ORDER OCTAL DIGIT ↓

00	02	04	06	10	12	14	16	
NUL	DLE	SP	0	@	P	`	p	0
SOH	DC1	!	1	A	Q	a	q	1
STX	DC2	"	2	B	R	b	r	2
ETX	DC3	#	3	C	S	c	s	3
EOT	DC4	$	4	D	T	d	t	4
ENQ	NAK	%	5	E	U	e	u	5
ACK	SYN	&	6	F	V	f	v	6
BEL	ETB	'	7	G	W	g	w	7

HIGH ORDER OCTAL DIGITS → / LOW ORDER OCTAL DIGIT ↓

01	03	05	07	11	13	15	17	
BS	CAN	(	8	H	X	h	x	0
HT	EM	)	9	I	Y	i	y	1
LF	SUB	*	:	J	Z	j	z	2
VT	ESC	+	;	K	[	k	{	3
FF	FS	,	<	L	\	l	¦	4
CR	GS	-	=	M	]	m	}	5
SO	RS	.	>	N	^	n	~	6
SI	US	/	?	O	_	o	DEL	7

Figure 2.



The first positions of sticks 4 and 6 are respectively the "commercial at" and "accent grave." Then the upper and lower case Roman alphabets follow. This offset of one position is historical (from the United Kingdom), and of no importance as long as you remember that it is so.

Following the alphabet in both sticks 5 and 7 are three positions each that one must be very cautious about. In ASCII they are assigned as [, /[1], and] in stick 5 — {, |[2], and } in stick 7. But in the ISO Code and CCITT versions they are reserved for national usage. Table II gives the national use assignment for these positions. Surely you remember that the Scandinavian alphabet has 29 letters, not 26? My friend Orjar Heen in Oslo is very protective of these positions. He says "If you Americans want to sell computers and software abroad, don't use the ASCII characters for these positions in your software."

To be more precise, positions 5/11, 5/12, 5/13, 7/11, 7/12, and 7/13 (noted above) are called *primary* national usage positions. So is 4/0, where ASCII has the "commercial at." Honeywell, for example, uses the "at" in its timesharing systems for deleting the previous character upon entry. But this isn't too serious, because many nations also have the "at" in their primary sets.

	currency		1st 7 national				dia	dia	1st 7 national			dia
	2/3	2/4	4/0	5/11	5/12	5/13	5/14	6/0	7/11	7/12	7/13	7/14
Netherlands—A			@	[	\	]	^	`	{	|	}	~
Australia	#		↓	↓	↓	↓	↓	↓	↓	↓	↓	~
Belgium—A	↓		↓	↓	↓	↓	↓	↓	↓	↓	↓	‾
W. Germany—A	↓	$	↓	↓	↓	↓	↓	↓	↓	↓	↓	‾
US	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	~
Japan	↓	↓	↓	↓	¥	↓	↓	↓	↓	↓	↓	‾
UK	£	↓	↓	↓	\	↓	↑	↓	↓	↓	↓	‾
Italy—A	#	¤	↓	↓	↓	↓	^	↓	↓	↓	↓	‾
Switzerland—A	↓	¤	↓	↓	↓	↓	^	↓	↓	↓	↓	~
France—A	↓	$		↓	↓	↓				↓		~
USSR	↓	¤		↓	V	↓	∧					‾
Netherlands—B					IJ					ij		
Belgium—B			à		ç				é	ij	è	‾
France—B	£	$	à		ç	§	^		é	ù	è	
Switzerland—B			à		ç				é	ù	è	-
Italy—B	#	$	§		ç	é	^	ù	à	è	è	ì
Switzerland—C					ç	é		ù	à	é	è	ì
Hungary	#	Ft	§	É	Ö	Ü	Á	ó	é	ö	ü	á
W. Germany—B	£	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ß
Switzerland—D				Ä	Ö	Ü			ä	ö	ü	ß
Sweden	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Finland				Ä	Ö	Å			ä	ö	å	
Denmark				Æ	Ø	Å			æ	ø	å	
Norway				Æ	Ø	Å			æ	ø	å	
Spain					Ñ					ñ		

Table 2.

Also in sticks 4-7 are three diacritical marks. They are accent grave (`) in 6/0, circumflex (^) in 5/14, and tilde (~) in 7/14. These are called *secondary* national usage positions. In some countries the tilde is a straight overline.

But it is the circumflex where we have a lot of confusion. Teletype first made it an "up arrow" in an earlier version of ASCII, to serve as an exponentiation symbol, primarily for BASIC. But that doesn't do very well, because the exponentiation for FORTRAN is a double asterisk! The FORTRAN version is preferable in France, certainly, because they use such words as crane, cote, cout, and so on.

A companion problem exists in position 5/15, with the underscore. The underscore is neither national nor diacritical; all countries use it just as underscore (and for typesetting it is a U.S. convention to indicate italics, but in Italy it means boldface, except when it is the last character in a line![3]). But Teletype's early version of ASCII used it as a "left arrow" — probably for an assignment symbol equivalent to := in ALGOL. The up and left arrow have been carried over from Teletype into many video terminals. Ask your terminal manufacturer to cease and desist and retrofit. It's not ASCII and will only cause trouble forever.

The last character in sticks 4-7 is the Delete, symbol DEL, in position 7/17. It was put here because the binary code is 1111111, which would be all punched holes in perforated (not always paper!) tape, and that is the only way to make sure that it cannot be misread as some other character. ASCII is a complete set; all positions are assigned to have meaning.

STICKS 2-3

These are usually called the sticks for digits and specials. Remember that they are the "digits" 0 to 9; not numbers, not numerals, not anything but digits! They are in 3/0 through 3/9 so that the low-order 4 bits are the representations for packed decimal. Originally we considered the possibility of a special 4-bit set for numerical applications (see the fifth entry in Table Ia), but it turned out that computer hardware became inexpensive enough to not deprive ourselves of the extra capabilities of the 7-bit and 8-bit sets.

		0000	0001	0010	0011	0100	0101	0110	0111
$b_8\,b_7\,b_6\,b_5$ / $b_4\,b_3\,b_2\,b_1$	COL / ROW	0	1	2	3	4	5	6	7
0000	0	NUL	DLE	SP	0	NOTE 1 @	P	NOTE 1 `	p
0001	1	SOH	DC1	!	1	A	Q	a	q
0010	2	STX	DC2	"	2	B	R	b	r
0011	3	ETX	DC3	NOTE 1 £/#	3	C	S	c	s
0100	4	EOT	DC4	NOTE 1 $	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y
1010	10	LF	SUB	*	:	J	Z	j	z
1011	11	VT	ESC	+	;	K	NOTE 1 [	k	NOTE 1 {
1100	12	FF	FS	,	<	L	NOTE 1 \	l	NOTE 1 ¦
1101	13	CR	GS	−	=	M	NOTE 1]	m	NOTE 1 }
1110	14	SO	RS	.	>	N	NOTE 1 ^	n	~ NOTE 1
1111	15	SI	US	/	?	O	_	o	DEL

Note 1

These 12 positions are variable for national usage — 2 for currency, 7 primary national usage, and 3 secondary usage which are diacritical marks when preceded by BSP. The presently-known assignments are given in the table below.

Figure 1.

Position 2/0 is officially called "space." I don't and didn't like it, and would have preferred "blank." Which is why the IBM community often uses a lower case "bee" with a slash through the vertical as its symbol. From the Univac side, the space has the official symbol "delta."

Having mentioned packed decimal, where two digits go into each 8-bit group ("byte" to the American, "octet" to the French), a word of caution on the plus and minus signs — they are in stick 2, rather than stick 3 with the digits. But the low order 4 bits are distinct, and + should be used only as 1011, − only as 1101. I mention this because the nonstandard code EBCDIC permits multiple representations of + and − in packed decimal. And the ASCII representations are not even coincident with any of these, with obvious dangers!

Watch out for the "currency" positions, 2/3 and 2/4. They also have national variations. In ASCII they are customarily # and $, but there are some things to be remembered:

- # is not "number sign" for many countries, most of which use "No." or "Nr." for that purpose. And when it is "number," it must precede the digits, not follow.
- # closely resembled the "sharp sign" in music.
- # is "pound sign" only for the U.S., the only major country still not using the metric system. To the rest, it's kilograms. For now, it's best to use the abbreviation "lb." in the U.S., not the #. In any case, both must *follow* the numeral.
- To the British, a "pound" has the symbol "£", which is why that is the symbol in position 2/3 for the UK. They get very irked when # is called a "pound" sign, especially in software manuals.
- The "dollar" is peculiar to the U.S., Canada, and some others. There are also francs, marks, escudos, pesos, lire, etc., etc. Which is why the ISO code uses the universal currency symbol in position 2/4. It's a circle with outside spikes at 45, 135, 225, and 315 degrees (¤), called "scarab." Table II also shows these assignments for several countries.
- ECMA has provided a separate guideline for specifying international currencies. See the "Where to Get More Information" at the end of this article.

It's a tough problem, and will get worse when we get into expanded character sets for photocomposition and such. For now, all we can do is follow the ASCII standard, which says that # is a "number sign."

Only a few more peculiarities remain for sticks 2-3. An important one is in the double quote, position 2/2, and the single quote, position 2/7. That is, you may think it is a single quote, and even use it so, but it is really an "accent acute" for vowels. It slants from top right to bottom left, to complement "accent grave" in 6/0, which slants from top left to bottom right. Some terminal makers do not realize this pairing, and will have accent grave slanting correctly, but put accent acute as a single quote in the unstylized up and down method. My Terminet is one of those that is OK.

Don't forget that to the typesetter, in contrast to typewriters, both single and double quotes have two forms — opening and closing. In fact, the typesetter gets his double quotes by using two single quotes, of either form, because the quote uses very little space in variable space typesetting. Most terminals, either video or hardcopy, use constant spacing. So double and single quotes must be distinct for that reason.

The last variation is in position 2/6, the ampersand. There are many legitimate different symbols for the ampersand. Neither ASCII nor the ISO Code prescribe any particular one. But this leads us to the next topic — how to represent the ASCII characters in handprinted form, so that they may be input to computer systems.

HANDPRINTING FOR STICKS 2-7

The classical confusion for many years was between the digit zero and the letter "oh," but there are other possibilities for confusion. American Standard X3.45 specifies the handwritten character shapes shown in Figure 3.

Figure 3.

This clears up a longstanding problem. The communications types, and the armed services, used to put a slash through the zero; somehow the IBM users got to putting the slash through the letter "oh" instead, confusing the Scandinavians greatly. Now it's neither (which helps), just a 180-degree rotation of the letter Q. The earlier German Standard DIN 66 002 prescribed the cursive loop in the upper right, as some may have learned in penmanship courses. It now permits the ANSI form as well.

UPPER AND LOWER CASE LETTERS

Many people are accustomed to using upper case only. This is a hangover from early line printers and limited sets (until the Stretch computer of IBM, characters were usually 6 bits in size). It would have been far better if they had all been lower case in those smaller sets. Putting it simply, would you buy a book to read if it were all

in upper case? Because lower case is much easier and faster to read, lower case should be the default case when one has only the one case. There is no reason why FORTRAN or BASIC processors cannot understand lower case variable names and verbs just as easily as they can understand upper case.

I always recommend getting a terminal with both cases if it is at all affordable. Second best is making sure that a single-case terminal is retrofittable later, if necessary. And if a single-case terminal, get it in lower case only, if possible. There has been much reportage in the computer trade press about eyestrain resulting from using computer terminals. Is the reason obvious?

STICKS 0, 1

These are the control characters. The most important distinction in ASCII is the split between sticks 0-1, Controls, and sticks 2-7, Graphics. We'll see this later on in the standards for Code Expansion (to 8 bits or more), and Code Extension (alternate sets, such as Cyrillic for the USSR, and Kata Kana for Japan).

Unfortunately, there is, despite the standard, much difference between the ways that various terminal devices handle these control characters. They may act differently, or they may not be operative at all. I have two very useful programs, written in the TEX language (Reference 6). One lists each symbol by name and then shows its action between parentheses. The other asks you to depress in turn all the funny keys on your terminal, and then tells you what control character(s) they generate, if any.

GRAPHICS FOR THE CONTROLS

There are standard graphical representations for the 32 controls, space, and delete. They are defined by ISO 2047, American Standard X3.32, and ECMA-17, and are shown integral to Figure 1. Some terminals are advertised as ASCII terminals, and yet generate Greek or other characters for these positions. Don't believe it! These symbols are every bit as useful as any Greek characters could be.

There are five groups in the basic control set.

STICKS 0, 1 — Logical Communication Control (10)

This group is used for both communication and for labeling of media. It includes:

SOH (0/1) (Start of Heading) — used as the first character in the heading of an information message.

STX (0/2) (Start of Text) — terminates the heading just before the text.

ETX (0/3) (End of Text) — Last character in the text message. Unfortunately, it is generated on many terminals via Control-C, and that's just to the right of Control-X on the keyboard, which is commonly used to cancel a bad input line. And if you mis-key — ouch!

EOT (0/4) (End of Transmission) — the last character in any transmission, and usually it turns your device off!

ENQ (0/5) (Enquiry) — requests a response from a remote station, either an identification of that stations (Who are you?) or its status.

ACK (0/6) (Acknowledge) — used by a receiver to reply "yes" to a sender.

DLE (1/0) (Data Link Escape) — an Escape character, especially for communications, analogous to ESC (1/11). It signals the start of a character sequence that causes a shifting into another set of communication controls, whenever they are needed.

NAK (1/5) (Negative Acknowledge) — used by a receiver to reply "no" to a sender.

SYN (1/6) (Synchronous Idle) — needed by synchronous transmission systems to get into, or stay in, *synchronization* when no other such signal is available to them.

ETB (1/7) (End of Transmission Block) — indicates the end of some division of data that the transmission system must make, unrelated to any division in the format of the logical data itself.

One lists each symbol by name and then shows its action between parentheses. The other asks you to depress in turn all the funny keys on your terminal . . .

STICKS 0, 1 — Physical Communication (4)

This group is used for communications. It includes:

NUL (0/1) (Null) — the standard says that it is "used" to accomplish media fill or time fill" . . . "may be inserted into or removed from a stream of data without affecting the information content of that stream." And that's exactly what the standard also says about DELete (7/15), which it lists as a control character even though it is not in the control sticks! The only difference I can see between them is that on perforated tape you can make any character into a DELete, but none into a Null.

CAN (1/8) (Cancel) — the receiver is to disregard the data received up to that point, starting from restart point that receiver and sender have agreed upon. It is common in timesharing for Cancel (often generated by a Control-X) to work on a line-at-a-time basis, to delete an unwanted string of entry characters, and effectively put one back to the position of re-entering the entire line. In this case, the agreement between sender and receiver is "back to the last CR." But there are many other ways that Cancel could be used, and for parallel as well as serial transmission.

SUB (1/10) (Substitute) — a character that says probably we would have had another character in this position if we could have figured out what it was supposed to be! There are many reasons for such confusion — perhaps parity didn't check out. But it is better to put in a SUB to keep the field lengths and such correct. Moreover, note its symbol, a mirror image (not the Spanish inverted) question mark. If this is displayable, it will tell you definitively that the system doesn't know what it is, and you can make a good guess in many cases, particularly in word text.

EM (1/9) (End of Medium) — defines the previous character as the last usable character on that medium, whether or not there is more recordable space on the medium.

STICKS 0, 1 — Device Control (11)

This group is used for control of devices such as terminals.

HT (0/9) (Horizontal Tabulation) — the standard says that is "advances" the active position to the next predetermined character position on the same line." There are two ways this can work:

1. Right at the terminal, if it has the horizontal tab capability built in. Sometimes you can set the tab positions by using the terminal only; almost always the computer can be made to set the tabs on the terminal. Then when you hit HT during entry, or HT is read from the computer output, the printing or displaying (active) position will skip to the next tab setting.
2. By a formatting program in the computer, which must be given some indication of the tab setting positions in force at any particular point in the file. The program then simulates horizontal tab movement by filling the lines with spaces as needed to achieve the alignment.

VT (0/11) (Vertical Tabulation) — the standard says that it "advances the active position to the same character position on the next predetermined line." And if you agree with somebody else, it can be to the first position in that line instead. This is a very dangerous character to use. It cannot be used directly on any terminal that I know of. Even if it could, the implementation rules are not supplied unambiguously in the ASCII standard. And for use by a formatting program, one would have to predefine the number of lines to be skipped. That's pretty tough when you are inserting and deleting lines, as every programmer knows.

LF (0/10) (Line Feed) — like vertical tab, but just to the next line, which is clean enough. If receiver and sender agree (again as in vertical tab), it can be to the first position of the next line, in which case it is called New Line (NL). Some manufacturers implement this. I personally prefer having a separate Carriage Return and Line Feed. Both codes can be generated with a single keystroke, and they often are.

FF (0/12) (Form Feed) — again like vertical tab, to the same cahracter position unless sender and receiver agree that it is to the first position in the new line, except that the tab is to a new line position that is related to a form of some size (those that fold 11 inches apart, for example). This control could run wild if your terminal or other display device is not equipped to handle it, so use it with caution in files.

CR (0/13) (Carriage Return) — moves the active position to the first position on the *same* line! Not like typewriters. They have effectively incorporated the New Line feature. But the non-advancing CR is better for terminals, even if it is misnamed. Neither video terminals nor ball and daisy wheel typewriters have carriages, so live with it.

BS (0/8) (Backspace) — Backspace is a very tricky character. On some terminals, such as video terminals, there is no key to generate Backspace for entry into the text stream or buffer. On many it can be created via Control-H. Even then, it may or may not be operative.

Backspace is meant for physical movement of the active position (which may or may not coincide with a cursor position, when such exists). Historically, it was included for hardcopy terminals and other hardcopy devices for some of these uses:

- Underscoring (underlining).
- Other forms of highlighting, such as bold. For example, the sequence A BS A BS A would strike the A three times on a hardcopy device, and make it look boldface (such a sequence can also be translated to call a boldface font in photocomposition).
- Editing indications. For example, in legislative bill drafting to indicate the deleted or changed portion:
  This is obsolete.
- Forming composite characters, e.g.:
  X̊ ± ≠ ↑ ⌶ ‖ } F (Hungarian forint)
- Forming accented letters, primarily for European languages. Examples:
  Ä Å Ø (Scandinavian letters following Z)
  Ñ a̍ à ô ü

Warning: Backspace is entirely different from a cursor movement on a video terminal! When the cursor is moved to a position where a character is already entered, succeeding entry in that position usually destroys the original character and replaces it with the new entry.

I personally haven't seen any video terminals with a true backspace. A former president of Infoton told me it could be done as an engineering special for about $5,000 one-time cost.

Warning: There are three ways to create underscored text for hardcopy terminals:

1. The characters, that many backspaces, and that many underscores (or vice versa).
2. A character, BS, underscore, the next character, etc. This is called the canonical form, and is used quite commonly.
3. Underscore, BS, character, underscore, etc.

I have noticed a lot of difficulty moving back and forth between hardcopy (at my home) and video (in my office) terminals. One tends to underscore on the hardcopy terminal and forget that half of the pairs are going to be wiped out by the cursor on the video terminal. In the first two methods above, it's the text that gets wiped out, and it's hard to read on the fly. So if you plan to display a file on a video terminal, find another highlighting method, or use the third underscoring convention. Even that may give problems if done by embedding an underscoring command in the file you pass to a formatting program; most such programs put the underscore last instead of first.

BEL (0/7) (Bell) — sounds an audible signal to get the user's attention. Some terminals are not so equipped, but they should be. It's good human engineering. But please give me an adjustable volume control!

And then there are the four device controls for unspecified purposes, DC1, DC2, DC3, and DC4 — in positions 1/1 through 1/4. Different manufacturers treat these like a wild card in poker — they make them anything that they want. Doesn't lead to much compatibility, so beware.

STICKS 0, 1 — Field Separators (4)

This group is used for formatting and string processing. These are the separators in positions 1/12 to 1/15. I got the idea originally from the Word Mark in the IBM 1401, which used an extra bit in the low-order character in a field as a delimiter. ASCII uses special and separate characters to indicate a hierarchical structure. Originally I put in eight such characters, but only these four remain:

FS(FileSeparator — 1/12)
GS(GroupSeparator — 1/13)
RS (Record Separator — 1/14)
US(UnitSeparator — 1/15)

FS is most inclusive, US the least inclusive. And we can consider the blank/space as the next lower order separator from these. Suppose we had a line of text like this:

(text1)US(text2)US(text3)RS(text4)US(text5)GS(text6)

On many terminals these delimiting control characters would not print, so we would see only a continuous stream. On others they might show as spaces. A TEX command to break the line at the record separator would be:

scan:line:*rs

The variable *left would contain "(text1) . . . (text3)". The variable *right would contain "(text4) . . . (text6)".

STICKS 0,1 — Changing Sets (3)

This group is used for moving to and from alternate graphic and control sets. This includes ESCape (1/11), Shift Out (0/14), and Shift In (0/15).

These basic control characters have permitted design of a quite marvelous structure for extension and expansion. It allows us to code and classify most of the world's graphic symbols for computer storage, interchange, and display. This big area will form most of Part III of this article.

IN THE NEXT INSTALLMENT

The ASCII Collating Sequence
ASCII and Programming Languages
ASCII and Media
Keyboards
ASCII and Display/Printing
Code Extension — Alternate Controls
Code Extension — Alternate Graphics
ASCII and Non-Latin Alphabets
Code Expansion — 8-bit ASCII

WHERE TO GET MORE INFORMATION

There are four sets of Information Processing Standards that may be of concern to you:

- ISO. Sold only through ANSI (American National Standards Institute), which has the franchise. That makes the prices high — much higher than in other countries.
- ANSI. These are American National Standards developed via the X3 and X4 committees, mostly. Prices still pretty high.
- ECMA (European Computer Manufacturers Association), 114 Rue du Rhone, 1204 Geneva, Switzerland). Free, and they have a lot more advanced standards than ISO and ANSI. But a modest donation would not be unwelcome.
- Your friendly U.S. Government, in the person of the Department of Commerce, National Bureau of Standards, Institute for Computer Sciences and Technology, in Gaithersburg, MD 20760. If by any chance you are employed by the U.S. Government, you get FIPS PUBS (Federal Information Processing Standards Publications) for cheap. Otherwise, see ANSI. (Refer to Tables 1a, 1b, and 1c). In many cases they are essentially reprints of the ANSI standards, for a fraction of the cost.

If you can't wait for the standards to be approved and published, catch them in progress. Ask CBEMA, the sponsor of ANSI X3, to put you on an observer list for the committee in your area of interest. The address is:

Robert Brown, Director of Standards
Computer & Business Equipment Manufacturers Association
1828 L Street NW
Washington, D.C. 20036
(202) 466-2288 Telex 89 29042

REFERENCES

1. ANS X3.4-1977, available from the American National Standards Institute, 1430 Broadway, New York, NY 10018.
2. ISO 646, avialable from ANSI (Reference 1).
3. R.W. Bemer, "ASCII — the data alphabet that will endure," in *Management of data elements in information processing*, National Bureau of Standards, 1975 October, 17-22.
4. R.W. Bemer, "A view of the history of the ISO character code," Honeywell Computer J. 6, No. 4, 1972, 274-282.
5. E.H. Clamons, "Character codes: who needs them?", Honeywell Computer J. 5, No. 3, 1971, 143-146.
6. The TEX Subsystem of the Timesharing System, Series 60 Level 66, Honeywell Information Systems, 200 Smith Street, Waltham, MA 02154, Order DF72.

ACKNOWLEDGEMENTS

Thanks go to co-workers at Honeywell Information Systems: Eric Clamons for much background, insight, and experience gained from working for a long time as chairman of X3J2 — the committee charged with the development of ASCII. And to Pat Skelly, ACM representative on ANSI X3, for collecting all the various national and international standards documentation upon which many of the figures were based.

FOOTNOTES

[1]For those curious about the reverse slash, it came from ALGOL 58. The reference language specified ∧ and V as the symbols for AND and OR respectively. I put the reverse slash in so these could be made as 2-character groups — and

[2]You will still see many terminals where this vertical bar is broken in the middle. This resulted from a hassle with the PL/I people, who wanted to stylize the exclamation point (2/1) as a vertical bar for OR in that language. And of course that would make the graphics the same. The compromise (at horrendous cost in people time) was to break the real vertical bar in ASCII. But it turned out that the PL/I people didn't really need it, or else it gained no momentum, so the real vertical bar is back to normal in ASCII-1977. Let's fix those terminals.

[3]The Italians also have a different solution to hyphenation and right justification. It ignores the syllable structure and simply demands that if, when you get to the last position in the line, the current word is not yet completed, that last character shall be underscored, and the word continued without fuss on the next line. I rather like it.

THE FATHER OF ASCII, Robert W. Bemer

Robert "Bob" Bemer received his A.B. in Mathematics from Albion College in 1940, and a Certificate in Aeronautical Engineering from Curtiss-Wright Tech a year later.

His vast work experience includes employment with the major leaders of the aircraft and electronics industries— most recently, as Senior Consulting Engineer with Honeywell Information Systems.

Highlights of his many accomplishments include: The discovery of polynomial telescoping (1954); creation of the PRINT 1 programming system for IBM (1956); development of FORTRANSIT; development of COMTRAN, one of the three major inputs to COBOL; development of XTRAN, predecessor to ALGOL (1958); was a major influence in the choice of the 8-bit character in IBM System 360 (1960); an influence in building the 1108 and 6000 systems; and editor of Honeywell Computer Journal.

He has an impressive list of over 71 publications to his credit.□

WHAT DO YOU REALLY NEED WHEN YOU BUY A COMPUTER SYSTEM?

When you buy a computer system from MISSION CONTROL you get three things: THE BEST PRICE, THE RIGHT COMPONENTS, AND THE SUPPORT TO GET IT UP AND RUNNING.

OUR SYSTEM PRICES ARE UNBEATABLE. We guarantee compatible components. Before you buy come by our store for "hands-on" experience on our demonstrators. Many people will try to save a few dollars by buying a piece here and a piece there, but when your system doesn't run who do you turn to?

At Mission Control we stand behind everything we sell with guidance and technical support. From our store you'll not only get what you want but also everything you need including peace of mind.

IMSAI 8080
22 slot, front panel, kit **$599**

APPLE II 16 KRAM
Color graphics **$1455**

SOROC IQ 120
Block mode **$995**

TI PROGRAMMER
Binary, hex converter **$49**

MICROPOLIS DUAL FLOPPY
630 K bytes **$1799**

NORTHSTAR FLOPPY
90 K bytes, kit **$599**

NORTHSTAR HORIZON II
Dual floppy, 32K RAM Soroc IQ 120. Basic Centronics 779 printer plus general business software **$4750**

ALPHA MICRO SYSTEMS
Power of a mini at the price of a micro

AM 100 —
2 board CPU **$1495**

AM 200
Floppy disk controller **$695**

WE THINK IT TAKES MORE THAN JUST HARDWARE TO MAKE A COMPUTER SYSTEM

MISSION CONTROL

2008 WILSHIRE BLVD.
SANTA MONICA, CA 90403
213 - 829-5137

T-Sat 11-6 Th 11-9

NAME ______________________

ADDRESS ______________________

CITY ______ STATE ______ ZIP ______

(______) ______________________
PHONE

☐ CHECK ☐ MONEY ORDER
☐ VISA ☐ MASTER CHARGE

CARD # ______________________

INTERBANK # ____________ EXP. DATE ________

item	price
tax	
total	

CA. RES. ADD 6% TAX. All prices cash. Credit cards add 3%. Shipping charges add $15 for computers, terminals, $10 for floppies, $5 for boards, $2 per calculator.

RCA CDP 1802 COSMAC MICROPROCESSOR

By Joseph Pallas

Microprocessor technology is growing by leaps and bounds. Sometimes it seems that these advancements are overlooked or just plain overshadowed by pre-existing hardware. Hopefully, this article will inspire many readers to take another look, to see what's available.

The RCA CDP 1802 microprocessor first reached the average hobbyist when Popular Electronics published plans for what is probably the least expensive complete microcomputer ever, the COSMAC Elf™. Since that time, the chip has been steadily rising in popularity. In addition to the "Elf," a number of other similar machines built on the 1802 have shown up, and a users group has been formed. Later, some of the products based on the 1802 will be discussed.

INTERNAL ORGANIZATION

Figure 1 shows the internal arrangement of registers, ALU, accumulator, and data bus in the system. It is important to note that the processor is oriented toward the hexadecimal system. The two 4-bit registers I and N hold the high and low HEX digits of the instruction, respectively. The N digit is used in some cases to specify one of the scratchpad registers. Th 8-bit D or data register is the accumulator, while DF (for data flag) is the carry bit.

The 4-bit X and P registers are each used to select one of the scratchpad registers, for the stack pointer/index and program counter, respectively. These can both be changed with single instructions, to simplify multiple stack locations and program counters. The T register stores the contents of these registers for interrupts and subroutines.

The Q register is one bit (i.e. a flip-flop) which can be set, reset, and tested, and is also brought out as an output. The interrupt enable bit determines whether interrupts will be recognized.

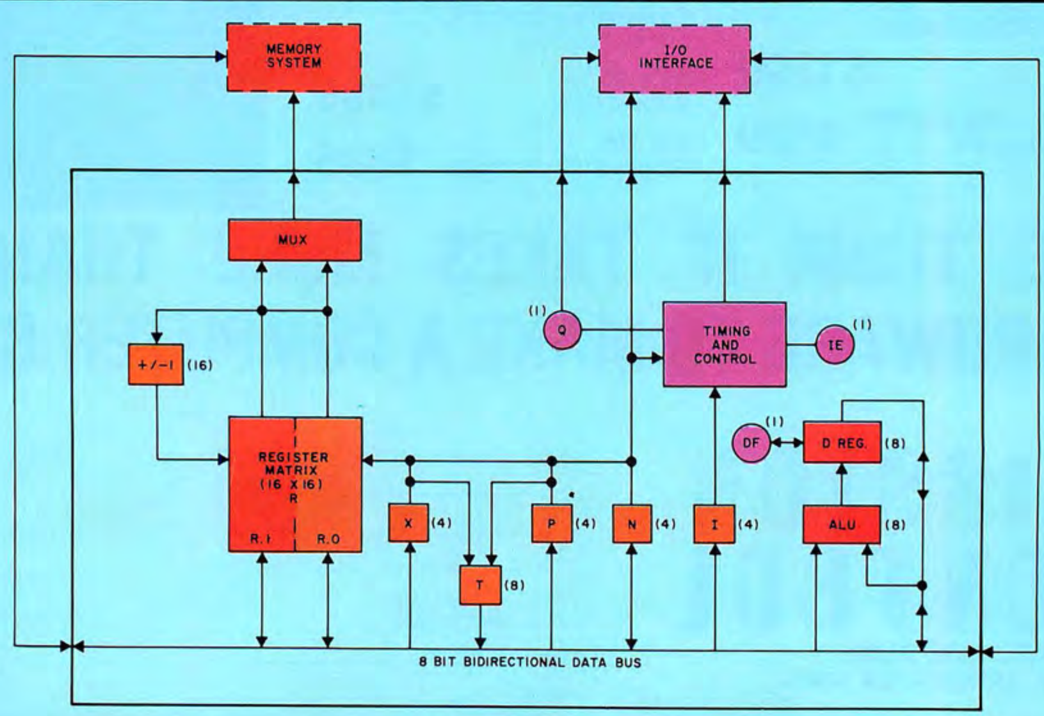


Figure 1. CDP 1802 Architecture.

There are 16 scratchpad registers, designated R(0) through R(F). Each of these holds two bytes, but *they are not 16-bit registers.* The two bytes are tied together only by the increment/decrement function. They can each be used as two 8-bit scratchpad registers or a 16-bit memory address register. Remember, *all* memory addressing is by these registers; there are no relative addressing modes.

The four external flags (EF1-4) can be tested by branch instructions. An excellent application for these is in I/O, where they can be used as "data ready" flags. An interrupt forces 1 into the P register, making R(1) the program counter. The DMA lines cause bytes to be loaded into or read from memory, with R(0) as the pointer.

The 16-bit addresses are *multiplexed* onto the eight address lines, they can be picked up in sync with the two timing pulses, TPA and TPB. Also, the timing pulses are used to catch outputs.

The WAIT and CLEAR lines give control over the processor itself. The CLEAR is equivalent to reset (X, P, Q = 0, interrupts enabled), and WAIT is equivalent to pause (used to single-stepping and slow memory/peripherals). Asserting both, forces the CPU into the load mode. Bytes can be loaded into memory via the on-chip DMA in this mode.

INSTRUCTION SET

Register operations. The following instructions have as operands one of the scratchpad registers: INC N, DEC N, IRX (increment the register currently designated as index/stack pointer, GLO N (load D with the lower byte of register N), PLO N (store D in register), GHI N, PHI N (G for GET, P for PUT).

Memory reference. LDN (load D with byte addressed by scratchpad register N), LDA N (LDN, then INC N), LDX, LDXA (POP), LDI #XX (load immediate), STR N (store D at address in register N), STXD (PUSH).

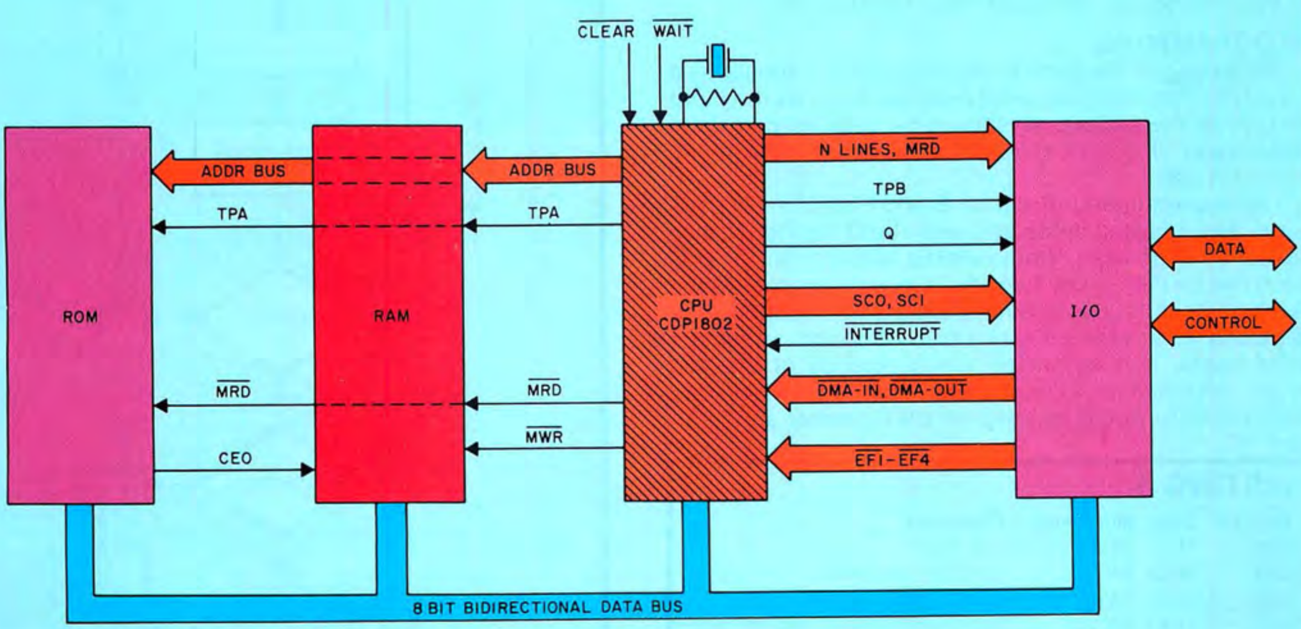


Figure 2. Typical CDP 1802-based microcomputer system.

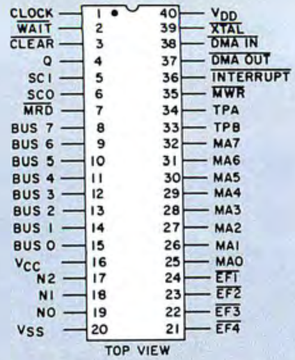


Figure 3.CDP 1802 COSMAC.

ALU. The ALU instructions operate on the D register and either an immediate byte or the byte addressed by the index/stack pointer (except the SHIFT's, of course): OR, ORI, XOR, XRI, AND, ANI, SHR (rightmost bit into DF, leftmost bit = 0), SHRC (rightmost bit into DF, DF becomes leftmost bit), SHL, SHLD, ADD (DF = carry), ADI, ADC, ADCI, SD (subtract D from memory, DF = borrow), SDI, SDB, SDBI, SM (subtract memory from D), SMI, SMB, SMBI. Note that the shift instructions simulate a 9-bit shift register.

Branch. The following are "short branches," branches within the current page: BR, BZ (if D = 0), BNZ, BDF, BNF, BQ (if Q = 1), BNQ, B1 (if Ef1 = 1), BN1, B2, BN2, B3, BN3, B4, BN4. These are "long branches" to any point in 64K of memory: LBR (hi-byte, lo-byte), LBZ, LBNZ, LBDF, LBNF, LBQ, LBNQ. SKP (skip next byte), LSKP (skip next two bytes), LSZ, LSNZ, LSDF, LSNF, LSQ, LSNQ, LSIE (if interrupt enable = 1).

Control. IDL (wait for DMA or interrupt), NOP, SEP (designate a register as PC), SEX (designate a register as index/stack pointer), SEQ (set Q), REQ, SAV (PUSH T register), MARK (PUSH X, P, move X, P to T, set X = P), RET (POP X, P; enable interrupts), DIS (same, disable interrupts).

I/O. 61 - 67 = OUT 1 - 7, 69 - 6F = INP 1 - 7, 68 is unimplemented. INP N (store byte from device on stack and in D), OUT N (put byte addressed by index/stack pointer onto bus for output, then increment the register).

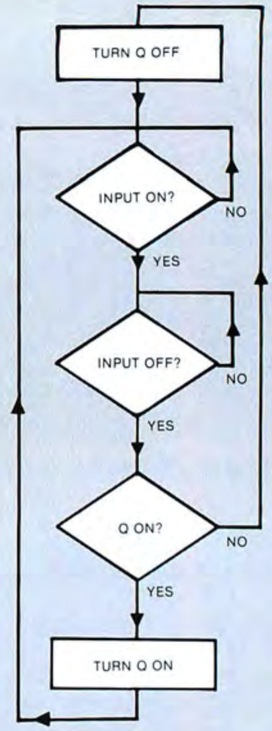
PROGRAMMING

The listing in Program A and Flowchart 1 simulates a "divide by two" flip-flop, with external flag 1 as the input and Q as the output. It should be relatively easy to understand, if it isn't, re-read the description of the instruction set.

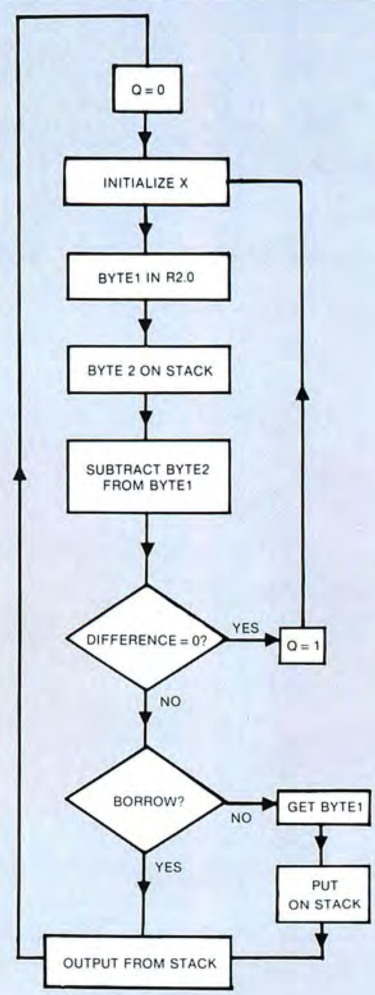
The second listing (Program B and Flowchart 2) compares two inputted bytes and sets the Q flip-flop if they are equal. Otherwise, it outputs the larger byte. The program has no data ready flags built in, so it processes the two bytes continuously. Similar programs might be used in closed loop feedback situations, although it might be more useful if it outputted the difference of the two bytes. Bear in mind, in most cases the first two registers (R(0) and R(1)) would be used as DMA pointer and interrupt PC.

LISTING 1

Address	Code	Mnemonic	Comments
0000	7A	REQ	turn Q off
0001	3C 01	BN1 01	wait for input pulse
0003	34 03	B1 03	wait for end of pulse
0005	31 00	BQ 00	if Q is on, turn Q off and start again
0007	7B	SEQ	else turn Q on
0008	30 01	BR 01	start again
			END



FLOWCHART 1



FLOWCHART 2

Seeing Is Believing

Khalsa Computer Systems, Inc.
A Professional Approach

Everyone can benefit from owning a computer, especially the business person. But how can you find the computer that fits your needs? You can shop around, or you can come to Khalsa Computer Systems, Inc. We are a dealership for many manufacturers. We select and stock what we believe to be the finest computer systems in every low cost category. Our staff is ready to discuss your exact requirements, and help you sort out all the alternatives, including leasing arrangements.

Our showroom offers you the opportunity to see the various computer systems in a calm and comfortable environment where the emphasis is on effective demonstration and education. Our friendly staff is trained to help you evaluate and configure computer systems that will solve your immediate problems and still allow for future expansion. In our showroom we provide separate booths for each computer system, plus reference books, periodicals, a classroom, lounge, and an office oriented demonstration area.

The selection of computers on display at Khalsa Computer Systems, Inc. represents the best low cost systems available regardless of manufacturer. We've done the research that guarantees you top quality hardware and software based on up-to-date technology. Our systems offer you low cost entry at four figures, yet give you more capability and flexibility than many larger systems offer for tens of thousands of dollars more. For instance, we can supply a complete multi-user timesharing system for under $10,000!

Our software group consists of experienced systems analysts and programmers who know how to solve your data processing problems. We provide services ranging from simple modifications of existing software to complete design and implementation of major software systems.

Our service department offers advanced, efficient service to meet the needs of our clients. We are proud of our new expanded shop and the skilled technicians who can answer the most technical questions about the equipment we sell. They are a vital part of the after sales support offered by Khalsa Computer Systems, Inc.

Come See Us

SYSTEMS BASED ON: Alpha Micro System AM 100, Poly 8813, Industrial Microsystems, Cromemco Z2D System 1, Processor Technology Sol 20, Imsai 8080, 8085 Systems, Vector Graphic, TEI. **DISC SYSTEMS:** PerSci, Calcomp Trident 50 MB-300MB, Shugart, iCOM, Micropolis, North Star, Industrial Microsystems, Cromemco. **PRINTERS AND TELEPRINTERS:** Texas Instruments Model 810 & Silent 700, Teletype Model 40 & 43, Centronics Model 761 & 799, Diablo 1620 & 1610, Microdata Matrix. **CRT TERMINALS:** Lear Siegler ADM 3A, Hazeltine 1500, Adds Regent 100 & 200, Soroc IQ-120, Textronics Graphic & Alphanumeric. **MEMORY BOARDS:** Industrial Microsystems 8K, 16K, Dynabyte, Processor Tech 16K, Extensys 64K, Ithaca Audio 8K Bare Boards. **SOFTWARE:** Microsoft Basic 4.4; Fortran, Digital Research CP/M & MAC, Structured Systems QSort General Ledger, Shrayer: Electric Pencil, BSL Vol. I-VII, Osborne Basic Prog. Discs, CP/M User's Group Collection.

KHALSA

COMPUTER SYSTEMS INCORPORATED

500 South Lake Avenue, Pasadena (213) 684-3311

Tuesday - Friday 12 noon to 8 pm, Saturday 12 noon to 6 pm, Closed Sunday and Monday

Khalsa Computer Systems, Inc. has also been known as the Byte Shop of Pasadena since we opened in 1976

LISTING 2

Address	Code	Mnemonic	Comments
0000	7A	REQ	clear Q
0001	E1	SEX R1	R1 = stack pointer
0002	F8 00	LDI 00	hi byte of stack address
0004	B1	PHI R1	
0005	F8 1A	LDI 1A	lo byte of stack address
0007	A1	PLO R1	
0008	69	INP 1	get first byte
0009	A2	PLO R2	put in R(2).0
000A	6A	INP 2	get second byte
000B	82	GLO R2	retrieve first byte
000C	F7	SM	first byte - 2nd byte
000D	32 13	BZ 14	if equal branch to 0014
000F	33 16	BDF 17	if positive answer (byte2 < byte1) go to 0017
0011	61	OUT 1	output from stack
0012	30 00	BR 00	start over
0014	7B	SEQ	activate Q
0015	30 01	BR 01	start over
0017	82	GLO R2	retrieve first byte
0018	51	STR R1	put on stack
0019	30 11	BR 11	output and start over
001A		STORAGE	stack location
			END

SUBROUTINES

There are three subroutine techniques outlined in the COSMAC User Manual. The first of these, called the "SEP Register Technique," is fast and simple, but places restrictions on the program's complexity. It reserves one register for each subroutine, and the subroutine must know the program counter of the calling segment. The subroutine is called with a 'SEP RN' instruction (after that register has been loaded with the correct address), and returns the same way. To restore the subroutine pointer, the subroutine returns from the point just preceding the subroutine's entry point. This technique is obviously limited, but it is useful because of the one-byte instruction, especially in subroutines with critical timing.

The "MARK Subroutine Technique" solves the problems with the MARK and RET instructions. The MARK instruction is used to store the values of X and P, which are restored by the RET instruction.

The "Standard Call and Return Technique" uses two subroutine 'handlers' which are called by SEP R4 and SEP R5 for call and return, respectively. The program and subroutines use R3 as the program counter. 'In-line' data from the main program can be picked up by the subroutine with one-byte load instructions, and separate stacks can be maintained for subroutines and data. For additional information and listings, consult the use manual.

INTERRUPTS

The 1802 does not have vectored interrupts, but they can be easily simulated with a little hardware and some software. The interrupt causes R1 to become the program counter, R2 to become the index/stack pointer, stores the old X and P in the T register, and disables further interrupts. Executing a SAV instruction and a STR R2 after a DEC R2 will save the state of the machine to enable the interrupt program to run. Returning from the interrupt is accomplished by restoring the D register (and any other registers saved) and executing either a RET or a DIS instruction. The effect is identical, except that DIS does not allow further interrupts.

SOFTWARE AND PRODUCTS

RCA does have a monitor (actually a debugger), an assembler, an editor, and a "utility program," actually a simple monitor, for the 1802. Some of these can be very expensive. Write to RCA for more information. Another item of interest is the "binary arithmetic subroutines" package. These *integer* routines add, subtract, multiply, divide, and do BCD/binary conversions. Diskette, paper tape, or cassette costs $100, but the manual, which costs $5, contains the source and object codes. It is possible to construct floating-point routines using these. One thing worth noting: it is fairly easy to rewrite 8080 software if the listings are well documented.

If you prefer to buy your software, RCA has funded the development of COSMAC Tiny BASIC by Itty Bitty Computers. This version is identical to the versions for 6800 and 6502 from the user's view, occupies the same 2K, and costs the same incredible $5. A more advanced BASIC is being developed by Infinite, which will include DATA, arrays, exponentiations, and loops. As always, write for more information.

RCA offers an "evalutation kit" which is suitable for experimenters. At $249, it offers the processor, an I/O port, a ROM containing the utility program, LED's on all data, address and control lines, and room for up to 4K of RAM. For another $140 you can get the Microterminal, which looks suspiciously like the SC/MP keyboard kit.

Infinite Incorporated offers, in addition to software, their UC1800, a complete unit with HEX keyboard and 6-digit HEX display. Also available is a set of bare PC Boards, some hardware (as in nuts & bolts), and keyboard encoding ROM. Quest Electronics offers a set of parts with power supply and PC board. The July 1977 Popular Electronics mentions the Elf II from Netronics. This kit includes PC board, HEX keyboard, parts, and room for expansion, and incorporates the CDP 1861 one-chip video interface for graphics. This single chip will map 512 bytes of memory with resolution comparable to that of the SWTPC GT-61 graphics terminal for less than $20. Of course, the system won't work with other MPU's.

CONCLUSION

Hopefully, the reader now has a better understanding of the organization and structure of the RCA CDP 1802 COSMAC microprocessor, and an idea of how he can use it himself in a variety of systems ranging from a complete BASIC computer with graphics to simple control applications. Examine the bibliography/reading list for other sources of information.□

BIBLIOGRAPHY/READING LIST

RCA:

MPM-109 COSMAC Microtutor Manual

MPM-201A User Manual for the CDP 1802 COSMAC Microprocessor

MPM-203A Evaluation Kit Manual for the COSMAC Microprocessor

MPM-206 Binary Arithmetic Subroutines for COSMAC Microprocessor

Application Notes: RCA #ICAN-6488, -6490, -6416, -6509, -6485, -6525, -6487, -6486.

MPM-920 Instruction Summary for CDP1802 (pocket-size card)

Built the COSMAC Elf," Part 1, Popular Electronics, August 1976; Part 2, September 1976; Part 3, March 1977; Part 4, July 1977.

An Introduction to Microcomputers, Volume II, Osborne and Associates, Inc., pp. 10-1 to 10-26.

SOURCES

RCA Solid State Division, Box 3200, Route 202, Somerville, NJ 08876.

Itty Bitty Computers, P.O. Box 23189, San Jose, CA 95153.

Infinite Incorporates, 1924 Waverly Place, Melbourne, FL 32901.

Quest Electronics, P.O. Box 4430, Santa Clara, CA 95054.

Netronics, 333 Litchfield Road, New Milford, CT 06776.

When you buy a microcomputer, is this your technical support?

A stack of do-it-yourself books?

At The Computer Mart, we back what we sell with technical support. From assembling kits for personal use to building microprocessor systems for business. Like word processing, simulation, control or data acquisition.

Don't kid yourself. Assembling a microcomputer can be a headache. Sure, kits may be inexpensive. But, they're also complex. It makes more sense to have the professionals assemble it. Professionals with experience gained from building hundreds of systems. It's the smart way to get your microcomputer up and running.

Our support doesn't end there. We stand behind you with a complete service department. We offer free counseling services so that you end up with the right combination of hardware and peripherals. We even have classes in software and programming. Plus an extensive library of books and periodicals to help you along.

At The Computer Mart, we support what we sell. So, stop in or call today. Take some of the headaches out of microcomputers.

THE COMPUTER MART

633 West Katella Avenue, Orange,
California 92667, (714) 633-1222

Hours Monday–Friday 10:00–8:00
Saturday 10:00–5:30

BankAmericard and Master Charge Accepted
Financing & Leasing Available

At The Computer Mart, we do more than just sell computers.

For professional advice before choosing your system, call The Computer Doctor.

A Development-Project

By William Rosenbluth
IBM Federal Systems Division

INTRODUCTION

When managing a significant development project, a plan is usually required so that all phases of the project may proceed in a correct sequence and so that the proper priorities may be placed on specific task components.

Whatever management technique used, the plan must provide for a logical flow of design, implementation, testing, and release tasks, showing sequential dependencies, degrees of completion, permissible overlap, critical paths, milestones, and critical items for special management attention. The complete plan must also show internal and external dependencies, expected services and support, and financial cost of execution.

The creation of the base activity network can utilize classic PERT-CPM[1-3] methods or use alternate techniques such as Mark III[4] or WOPAST[5]. Such methods and techniques are widely applied, but the reporting and use of status information is widely misused. Often this misuse permits unchecked overruns or else clouds fundamental issues in superabundant detail and rationalization so as to mask missed objectives.

The report and control scheme described herein gives the reader a working model with which he may run, control, or re-acquire control of, his project.

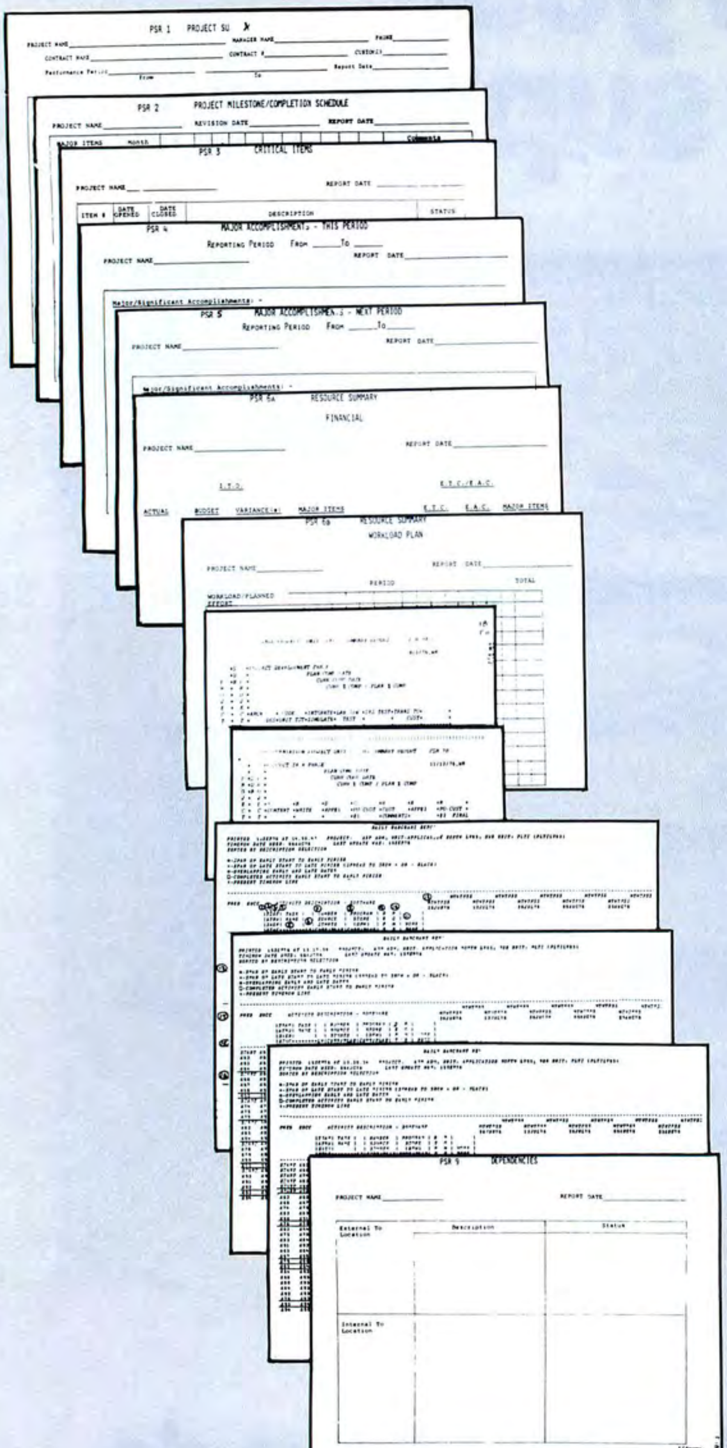


Figure 1.

OBSERVING & ASSESSING (See Figure 1, Ensemble of Project Status Reports)

Assuming activity is under way, or is planned via some network method, the manager must check point the status of his project in such a way that he can determine progress, plus or minus, at his next check point and report those facts to his superiors. The nine report types described herein and the use of those reports in a review hierarchy, allow him to do just that.

PROJECT STATUS REPORT (PSR) FORMATS

Status reviews are usually held for various levels of management — from first through n^{th}. The following summary reports cover almost all information required.

- Project Summary (PSR 1)
- Project Deliverable/Milestone Schedule (PSR 2)
- Critical Items (PSR 3)
- Major Accomplishments — This Period (PSR 4)
- Major Accomplishments — Next Period (PSR 5)
- Resource Summary $, Workload (PSR 6)
- Project Unit Status Summary Report (PSR 7)
- Sub-Unit Networks (PSR 8)
- Dependencies (PSR 9)

PSR 1 PROJECT SUMMARY (See PSR-1)

The project summary, PSR-1, contains the identifiers which commonly locate the project in a large organization. As indicated on the form they are:

Project Name — Common name of project (sub-portion of contract) or sub-project.

Contract Name & No. — Business identifier and legal-business reference for project.

Customer — Usually simple, but in many instances identifies varied versions of same project.

Manager Name, Phone — as indicated.

Reporting & Control Technique

Performance Period — as indicated, usually keyed to contract reviews, etc.

Report Date — Data of status review.

Brief Description of Project — as indicated.

Status — snapshot of project vital status showing record of last 10 reports.

Schedule — as indicated
Budget — u = under or ahead, %
y = on plan, ±5%
0 = over or behind, %
Significant Milestones — achievement dates per report
Estimated Completion — Project completion date
Major Problems — as necessary
Budget Status
EAC — Estimate of Expenditures At Completion
BTD — Budget To Date
ATD — Actual To Date

The Project Summary is designed to give Status-at-a-Glance for the busy executive who is not always able to participate in the full review.

PSR 2 PROJECT MILESTONES/COMPLETION SCHEDULE (See PSR-2)

The Project Milestone(s)/Completion Schedule is completed as follows:

Project Name — Common name of project or sub-project

Revision Date — Date of last reported schedule revision (on PSR-1). A program operating against the original schedules is indicated by writing "Original" in the Revision Date.

Report Date — Date of status review.

Major Items — List of all major sub-projects (tasks) for a 12-month period. Indicates by a bar graph (Gantt chart) the period of each task/phase. Each activity within the task/phase is to be identified on the bar. Use symbols as indicated at bottom of PSR-2 for highlights.

The next immediate 12-month performance schedule is reported by month.

Multi-year projects report a quarterly schedule for activity beyond 12 months. At completion of the indicated 12 months of schedule a new PSR-2 is prepared

PROJECT NAME______ MANAGER NAME______ PHONE______

CONTRACT NAME______ CONTRACT #______ CUSTOMER______

Performance Period______ ______ Report Date______
From To

BRIEF DESCRIPTION OF PROJECT

STATUS

______ Schedule
______ Budget

	Original (Date)	Report 1 (Date)	Report 2 (Date)	Report 3 (Date)	Report 4 (Date)	Report 5 (Date)	Report 6 (Date)	Report 7 (Date)	Report 8 (Date)
Significant Milestones									
1									
2									
3									
Estimated Completion									
Major Problems									
Budget Status									
EAC									
BTD									
ATD									

PSR 1 Project Summary

PSR 2 Project Milestone/Completion Schedule

PROJECT NAME____________ REVISION DATE____________ REPORT DATE____________

MAJOR ITEMS Month													Comments

PSR 3 Critical Items

PROJECT NAME____________ REPORT DATE ____________

ITEM #	DATE OPENED	DATE CLOSED	DESCRIPTION	STATUS

beginning with month 12 (previous) activity to provide carryover schedule information.

Guidelines:

- Major Items must list all outputs (original and added) and include important events (milestones) related to each listed major task. Use an additional sheet if necessary.
- An event is defined as a point in time at which some task(s) are accomplished. An activity is that work-effort which must be accomplished to finish a task.
- An event is generally made up of two or more task completions. A group of events comprise specific milestones.
- The symbology at the bottom of PSR-2 is used for uniform reporting notation.

PSR 3 CRITICAL ITEMS (See PSR-3)

Critical Items are completed as follows:

Project Name — Common name of project or sub-project

Report Date — Date of status review

Item # — Sequential number assigned to "Critical Item" for logging and tracking purposes.

Date Opened — Date "Critical Item" first reported

Date Closed — Date "Critical Item" resolved and removed from active log.

Description — Brief meaningful explanation of problem and planned action.

Status — Condition report on open "Critical Item"

Closed "Critical Items" are removed from periodic reports at the period after the item has been closed. Thus closed "Critical Items" are always known to reviewing management.

PSR 4 MAJOR ACCOMPLISHMENTS — THIS PERIOD (See PSR-4)

Major Accomplishments are completed as follows:

Project Name — Common name of project or sub-project

Report Date — Date of status review

Reporting Period — Period covered by this status review

Major/Significant Accomplishments — Brief meaningful explanation of accomplishment and significance to overall project, sub-project, or milestone.

PSR 5 MAJOR ACCOMPLISHMENTS — NEXT PERIOD (See PSR-5)

Same as PSR 4 except for period covered.

PSR 6A RESOURCE SUMMARY — FINANCIAL (See PSR-6A)

The financial resource summary indicates at-a-glance budget targets and variances for the project or sub-project. Items shown are:

Project Name — Common name of project

Report Date — Date of status review

I.T.D. — Initial-to-date expenditure summary

Actual — Funds expended to-date

Budget — Budgeted funds projected for expense to-date

Variance(s) — Funding deltas explained by major-item

Major-Items — Commentable reasons or sub-portions to explain variance(s)

E.A.C. — Estimate At Completion expenditures

E.T.C. — Estimate to Complete major item sub-portions of project

PSR 4 Major Accomplishments — This Period

REPORTING PERIOD FROM ______ TO ______

PROJECT NAME ______________________ REPORT DATE ______________________

Major/Significant Accomplishments: -

E.A.C. (TOT) — Total of I.T.D. (actual) ± E.T.C. major-items

E.A.C. (Budget) — Budgeted funds projected for project completion

E.A.C. (Variance) — Difference between E.A.C. (TOT) and E.A.C. (Budget)

Major-Items — as above.

PSR 6B RESOURCE SUMMARY — WORKLOAD PLAN (See PSR-6B)

The workload resource summary indicates period at-a-glance workload compared to planned, available, or projected effort. A review of period and term totals quickly indicate situations needing management attention. Items shown are:

Project Name — Common name of project

Report Date — Date of status review

Period — Usually months, calendar planning unit used to be consistent with financial and milestone summaries.

Workload — Management estimate of work-effort to complete project(s).

Planned Effort — (also available effort, projected effort, etc.). Manpower available to apply to project.

Major Items — Identifiable tasks or sub-projects

Total — Workload/effort total by period and by Major Item

PSR 7 PROJECT/SUB-PROJECT NETWORK SUMMARY REPORTS (See PSR's 7A & 7B)

Each project or sub-project of more than a modestly complex nature is generally tracked by a manually or computer generated activity network. One form of network is described in PSR 8. For a status review, all networks must be reduced to simple status summaries.

For the general case the project is characterized by task-phases, each representing a particular work sequence for that type of project or sub-project, and each having defined exit criteria for each task-phase.

Two examples should explain the above concept; Example 1 PSR 7A and Example 2 PSR 7B.

The included task-phase detail of the report is left to the responsible manager. It is common to report on key task-phases per project as is shown in examples PSR 7A (Software) and PSR 7B (Documentation).

For the generalized PSR 7 the following items are always shown:

Project Name — Common name of project

Project Date — Date of status review

Task-Phase — Explained in Example 1

Plan Comp Date — Original completion date of project at start of activity

Curr Comp Date — Current completion date as of the status review

Curr % Comp/Plan % Comp — Current progress (% of total work) with respect to original planned progress at this point in schedule.

Summary — The latest completion date for all sub-projects in the task phase is usually shown. The % completions are usually averaged over all sub-projects. See examples PSR 7A and PSR 7B.

EXAMPLE 1 PSR 7A
Software Development Task-Phases (Examples)

Major development task-phases define the various stages of software creation and modification. The following codes show an example of those task-phases.

A = Architecture and Design
C = Code and Unit Test

PSR 5 Major Accomplishments — Next Period

REPORTING PERIOD FROM ______ TO ______

PROJECT NAME ______________________ REPORT DATE ______________________

Major/Significant Accomplishments: —

PSR 6A Resource Summary
Financial

PROJECT NAME________________________ REPORT DATE________________________

I.T.D.				E.T.C./E.A.C.			
ACTUAL	BUDGET	VARIANCE(s)	MAJOR ITEMS	E.T.C.	E.A.C.	MAJOR ITEMS	
		______	______	____	____	______	
		______	______	____	____	______	
		______	______	____	____	______	
		______	______	____	____	______	
		______	______ TOTAL	____	____	______	TOTAL
					____		BUDGET
					____		VARIANCE

PSR 6B Resource Sumary
Workload Plan

PROJECT NAME____________________ REPORT DATE________________________

WORKLOAD/PLANNED EFFORT	PERIOD																TOTAL	
TOTAL																		

I = Integration and Simulation
L = Lab Software Test
S = System Test
T = Transmittal to Customer

Task-Phase	Description	Responsible Department
A	Architecture and Design encompasses the creation and mapping of software logical flow as well as definition of required interfaces, program utilities, and conventions.	
C	In the Coding and Unit Test Phase, individual program modules are implemented in the specific computer language specified and those modules are checked for functionability against unit-test procedures.	
I	Integration and Simulation combines software modules together as segments (of a final system tape) and functionally exercise them against a software simulator. After successful completion of Integration and Simulation, the application software is formally transmitted to the Lab Software Test Library.	
L	Lab Software Test includes the combination of all segments into the build of System Tapes and the exercise of these tapes on computer hardware. Signal-processing-mode tests are then run to confirm correct full-system processing.	
S	System Test responsibilities include the verification of the qualitative signal processing capabilities of the complete hardware-software-microcode system. System tape builds are tested against the development hardware and all other software (Diagnostics, etc.). Both software completeness and functionality are verified.	
T	Transmittal to Customer is the Program Office activity to release a computer tape to the customer.	

EXAMPLE 2 PSR 7B

Fourteen task-phases define the various stages of Documentation development. The following codes identify those task-phases in reports and status-summaries.

A FORMAT/CONTENT
B WRITE
C TYPE/REPRO 1
D APPROVAL 1
E TYPE/REPRO 2
F XMIT TO P.O.
G P.O. TO CUST
H CUST COMMENTS BACK
I INCORP COMMENTS
J TYPE/REPRO 3
K APPROVAL 2
L TYPE/REPRO 4
M XMIT TO P.O.
N P.O. TO CUST

PSR 8 PROJECT & SUB-PROJECT NETWORKS

There are many automated[1, 2, 3, 4] and surprisingly efficient manual[5] tools for the creation of sequential-dependency networks.

Illustrated here is one simplified example using a low-cost (less than $200/month rental), commonly available (remote-access APL[6]), network calculator called APL MINIPERT, available from the IBM Corporation (IBM Program Product 5734-XP3).

MINIPERT Networks for Software and Documentation Development

Shown here are the construction and use of tailored MINIPERT reports, as would be used to manage various software and documentation development projects.

A task-phased software development network is shown in PSR 8A. Pertinent features of PSR 8A are as follows:

1. **Task-Phase.** Key identifiers refer to the task-phases such as defined in PSR 7A and PSR 7B. These task-phases are used for primary sorts in determining work accomplished by task-phase.
2. **Status.** Completion percentage (X 10) for the module task-phase is indicated. Status variances shown in paragraphs 13, 14, 15 and 16 show how completion indicators are compared with calendar-allocated time for task completion reports.
3. **Area.** Reference to major managerial area to which the task is assigned.
4. **Function.** Reference to sub-area within that area of software assignment.
5. **Task Name.** Reference to either documentation or software or other work items tracked by the development network.
6. **Level.** Refers to the engineering change identifier for that particular task.
7. **Source Statements.** For the software example of PSR 8C, we are tracking the number of source statements in program modules; current count and plan. This is one key parameter in verifying completion status.
8. **Program Store Size.** For the software example of PSR 8C we track the internal computer storage requirements per module. This constraint is generally critical and so its allocation by module becomes the key parameter in fitting the whole software project into a computer internal storage.
9. **Department.** Refers to department working on functional sub-area.
10. **Programmer.** Refers to the person actually implementing that task.
11. **Work Package.** Refers to the lowest element of work breakdown structure against which that task activity work is charged.
12. **Calendar.** Shows as a heading to reference chart activity line associated with each marginal task-phase effort.
13. **Example 1**
    Indicated here is a finished module task-phase (note 'F' in status column) whose program store size has exceeded plan (169/151).
14. **Example 2**
    Indicated here is a module task-phase whose calendar allotment is past due and whose indicated "Status" is shown as 0%. We are behind schedule, and this is a candidate for a "Red Flag Item."
15. **Example 3**

PSR 7A Software-Project Unit-Status Summary Report

```
*                                                         8/1/76,WR

   *S    *PROJECT DEVELOPMENT PHASE:
   *U    *                         PLAN COMP DATE
F  *B P  *                            CURR COMP DATE
R  *  R  *                               CURR % COMP / PLAN % COMP
O  *  O  *
J  *  J  *
E  *  E  *
C  *  C  *ARCH     * CODE    *INTGRATE*LAB S/W *SYS TEST*TRANS TO*        *
T  *  T  *     DES*UNIT TST*SIMULATE*  TEST   *        *    CUST*        *
---*-----*--------*--------*--------*--------*--------*--------*--------*
S  *     *06/15/76*07/01/76*07/15/76*08/10/76*        *        *        *
A  * A   *        *        *        *        *        *        *        *
T  *     *100/   %*100/   %*100/   %*20/20  %*        *        *        *
T  *-----*--------*--------*--------*--------*--------*--------*--------*
   *     *06/04/76*06/15/76*08/10/76*        *        *        *        *
C  * B   *        *        *        *        *        *        *        *
O  *     *100/   %*100/   %*100/   %*15/20  %*        *        *        *
N  *-----*--------*--------*--------*--------*--------*--------*--------*
T  *     *06/01/76*06/01/76*07/15/76*08/10/76*        *        *        *
R  * C   *        *        *        *        *        *        *        *
O  *     *100/   %*100/   %*100/   %*25/25  %*        *        *        *
L  *-----*--------*--------*--------*--------*--------*--------*--------*
L  *     *06/04/76*06/11/76*07/05/76*08/07/76*        *        *        *
E  * D   *        *06/20/76*07/10/76*08/22/76*        *        *        *
R A*     *100/   %*100/   %*100/   %*10/25  %*        *        *        *
  P*-----*--------*--------*--------*--------*--------*--------*--------*
  P*     *06/01/76*06/01/76*07/15/76*08/12/76*        *        *        *
   * E   *        *        *        *        *        *        *        *
  S*     *100/   %*100/   %*100/   %*25/30  %*        *        *        *
  O*-----*--------*--------*--------*--------*--------*--------*--------*
  F*     *06/01/76*06/01/76*07/15/76*08/15/76*        *        *        *
  T* F   *        *        *        *        *        *        *        *
  W*     *100/   %*100/   %*100/   %*30/30  %*        *        *        *
---*-----*--------*--------*--------*--------*--------*--------*--------*
---*-----*--------*--------*--------*--------*--------*--------*--------*
   *     *06/15/76*07/01/76*07/15/76*08/15/76*10/01/76*10/15/76*        *
SUMMARY  *        *        *        *08/22/76*10/10/76*10/22/76*        *
   *     *100/   %*100/   %*100/   %*20/25  %*        *        *        *
---*-----*--------*--------*--------*--------*--------*--------*--------*
---*-----*--------*--------*--------*--------*--------*--------*--------*
*                                                                       *
*                                                                       *
*                                                                       *
*                                                                       *
*                                                                       *
*                                                                       *
*                                                                       *
*                                                                       *
```

EXAMPLE SOFTWARE SUMMARY REPORT

PSR 7B Documentation-Project Unit-Status Summary Report

```
 *
   *    *PROJECT TASK-PHASE:                                       11/12/76,WR
   *    *                        PLAN COMP DATE
P *S P  *                            CURR COMP DATE
R *U R  *                                CURR % COMP / PLAN % COMP
O *B O  *
J *  J  *
E *  E  *A        *B        *D        *G        *H        *K        *N        *
C *  C  *CONTENT  *WRITE    *APP#1    *PO-CUST  *CUST     *APP#2    *PO-CUST  *
T *  T  *         *         *         *#1       *COMMENTS*          *#2  FINAL
---*-----*---------*---------*---------*---------*---------*---------*---------*
S  *     *10/15/76*10/29/76*11/05/76*11/17/76*12/06/76*01/07/77*01/17/77*
A  * A   *        *        *        *        *        *        *        *
T  *     *100/    *70/70   *        *        *        *        *        *
T  *-----*--------*--------*--------*--------*--------*--------*--------*
   *     *10/15/76*10/29/76*11/05/76*11/17/76*12/06/76*01/07/77*01/17/77*
M  * B   *        *        *        *        *        *        *        *
N  *     *100/    *65/70   *        *        *        *        *        *
T  *-----*--------*--------*--------*--------*--------*--------*--------*
   *     *10/15/76*10/29/76*11/05/76*11/17/76*12/06/76*01/07/77*01/17/77*
P D* C   *11/15/76*11/29/76*12/05/76*12/09/76*12/20/76*01/15/77*01/25/77*
A O*     *90/100  *10/70   *        *        *        *        *        *
N C*-----*--------*--------*--------*--------*--------*--------*--------*
E U*     *11/12/76*11/19/76*12/01/76*12/09/76*12/31/76*01/14/77*01/24/77*
L M* D   *        *        *        *        *        *        *        *
  E*     *100/100 *60/70   *        *        *        *        *        *
  N*-----*--------*--------*--------*--------*--------*--------*--------*
  T*     *01/07/77*01/21/77*02/01/77*02/10/77*02/28/77*03/23/77*03/27/77*
  A* E   *        *        *        *        *        *        *        *
  T*     *50/50   *        *        *        *        *        *        *
  I*-----*--------*--------*--------*--------*--------*--------*--------*
  O*     *12/10/76*12/17/76*01/07/77*01/17/77*02/07/77*12/25/77*03/06/76*
  N* F   *        *        *        *        *        *        *        *
   *     *100/70  *10/0    *        *        *        *        *        *
---*-----*--------*--------*--------*--------*--------*--------*--------*
---*-----*--------*--------*--------*--------*--------*--------*--------*
   *     *01/07/77*01/21/77*02/01/77*02/10/77*02/28/77*03/23/77*03/27/77*
SUMMARY  *        *        *        *        *        *        *        *
   *     *80/90   *20/20   *        *        *        *        *        *
---*-----*--------*--------*--------*--------*--------*--------*--------*
---*-----*--------*--------*--------*--------*--------*--------*--------*
*  *     *        *        *        *        *        *        *        *
*  *     *        *        *        *        *        *        *        *
*  *     *        *        *        *        *        *        *        *
*--*-----*--------*--------*--------*--------*--------*--------*--------*
*  *     *        *        *        *        *        *        *        *
*  *     *        *        *        *        *        *        *        *
*  *     *        *        *        *        *        *        *        *
*--*-----*--------*--------*--------*--------*--------*--------*--------*
```

EXAMPLE DOCUMENTATION SUMMARY REPORT

DAILY BARCHART REPORT

```
PRINTED  13SEP76 AT 14.50.47   PROJECT:   ASP ADM; UNIT:APPLICATION SOFTW LV03; SUB UNIT: PLTI (PLTILV03)
TIMENOW DATE USED: 06AUG76        LAST UPDATE WAS: 13SEP76
SORTED BY DESCRIPTION SELECTION

o-SPAN OF EARLY START TO EARLY FINISH
•-SPAN OF LATE START TO LATE FINISH (SPREAD TO SHOW + OR - SLACK)
●-OVERLAPPING EARLY AND LATE DATES
□-COMPLETED ACTIVITY EARLY START TO EARLY FINISH
+-PRESENT TIMENOW LINE

-----------------------------------------------------------------------------------------------------------
                                                              (12)     MTWTFSS      MTWTFSS      MTWTFSS      MTWTFSS      MTWTFSS
PRED  SUCC (1)(2)(3)(4)ACTIVITY DESCRIPTION - SOFTWARE        MTWTFSS      MTWTFSS      MTWTFSS      MTWTFSS      MTWTFSS
                          (7)            (8)       (9)  (10)  28JUN76      12JUL76      26JUL76      09AUG76      23AUG76
           |TSAF| TASK |   | NUMBER    | PROGRAM   | D  P |(11)    |
           |ATRU| NAME |(6)| SOURCE    |  STORE    | E  G |        |
           |SAEN| (5)  |   | STMNTS    |  (DPW)    | P  M | WORK   |
           |KTAC••••••••LV|CURR/PLAN|CURR/PLAN| T  R | PKGE   |
-----------------------------------------------------------------------------------------------------------
START A62   AFALALBBC1  CT:   35/  27:    9/    8:NE4JBD:V43120:
START A66   AFALALBC1X  CT:  492/ 500:  260/  256:NE4JBD:V43120:
START A74   AFALALLTFX  CT:  906/ 893:  613/  225:NE4JBD:V43120:
START A78   AFALALPL3X  CT:1107/1000:  521/  531:W28KNR:V43122:
START A82   AFALALPLSC  CT:  863/ 750:  486/  492:NE4JBD:V43120:
(13)START A90   AFALALTFPC  CT:  203/ 200:  169/  151:NE4JBD:V43120:
A62   A63   CFALALBBC1  CT:                               :V43133:
A66   A67   CFALALBC1X  CT:                               :V43133:
A74   A75   CFALALLTFX  CT:                               :V43133:
A78   A79   CFALALPL3X  CT:                               :V43135:
A82   A83   CFALALPLSC  CT:                               :V43133:
A90   A91   CFALALTFPC  CT:                               :V43133:
A63   A64   IFALALCCDC  CT:  629/ 629:  436/  436:NE4JBD:V43245:
A75   A76   IFALALLTFX  CT:                               :V43245:
(14)A79   A80   IOALALPL3X  CT:                               :V43247:
A63   A64   IOALALTCDC  CT:  412/ 402:  262/  262:NE3ROM:V43248:
A83   A84   I2ALALPLSC  CT:                               :V43245:
A91   A92   I2ALALTFPC  CT:                               :V43245:
A63   A64   I6ALALBBC1  CT:                               :V43245:
(15)A67   A68   I8ALALDC1X  CT:                               :V43245:
A64   A93   LOALALBBC1  CT:                               :V43301:
A68   A93   LOALALBC1X  CT:                               :V43301:
A80   A93   LOALALPL3X  CT:                               :V43301:
A84   A93   LOALALPLSC  CT:                               :V43301:
A92   A93   LOALALTFPC  CT:                               :V43301:
(16)A76   A93   L5ALALLTFX  CT:                               :V43301:
A93   A94   S-PLTI SYS TEST                               :V45001:
A94   A999  T-PLTI XMT CUST                               :V61000:

          (18) TOTALS:4647/4401:2756/2761
```

(17) (17) (17) (17)

PSR 8A Software Network (Sorted by Task Type)

DAILY BARCHART REPORT

```
PRINTED  15SEP76 AT 13.17.06   PROJECT:   ASP ADM; UNIT: APPLICATION SOFTW LV03; SUB UNIT: PLTI (PLTILV03)
TIMENOW DATE USED: 06AUG76        LAST UPDATE WAS: 13SEP76
SORTED BY DESCRIPTION SELECTION

o-SPAN OF EARLY START TO EARLY FINISH
•-SPAN OF LATE START TO LATE FINISH (SPREAD TO SHOW + OR - SLACK)
●-OVERLAPPING EARLY AND LATE DATES
□-COMPLETED ACTIVITY EARLY START TO EARLY FINISH
+-PRESENT TIMENOW LINE

-----------------------------------------------------------------------------------------------------------
                                                                   MTWTFSS      MTWTFSS      MTWTFSS      MTWTFSS      MTWTFSS
PRED  SUCC     ACTIVITY DESCRIPTION - SOFTWARE                MTWTFSS      MTWTFSS      MTWTFSS      MTWTFSS      MTWTFSS
                                                              28JUN76      12JUL76      26JUL76      09AUG76      23AUG76
           |TSAF| TASK |   | NUMBER    | PROGRAM   | D  P |        |
           |ATRU| NAME |   | SOURCE    |  STORE    | E  G |        |
           |SAEN|      |   | STMNTS    |  (DPW)    | P  M | WORK   |
           |KTAC••••••••LV|CURR/PLAN|CURR/PLAN| T  R | PKGE   |
-----------------------------------------------------------------------------------------------------------
START A62   AFALALBBC1  CT:   35/  27:    9/    8:NE4JBD:V43120:
A62   A63   CFALALBBC1  CT:                               :V43133:
A63   A64   I6ALALBBC1  CT:                               :V43245:
A64   A93   LO1LALBBC1  CT:                               :V43301:
START A66   AFALALBC1X  CT:  492/ 500:  260/  256:NE4JBD:V43120:
A66   A67   CFALALBC1X  CT:                               :V43133:
A67   A68   I8ALALDC1X  CT:                               :V43245:
A68   A93   LOALALBC1X  CT:                               :V43301:
A63   A64   IFALALCCDC  CT:  629/ 629:  436/  436:NE4JBD:V43245:
START A74   AFALALLTFX  CT:  906/ 893:  613/  225:NE4JBD:V43120:
A74   A75   CFALALLTFX  CT:                               :V43133:
A75   A76   IFALALLTFX  CT:                               :V43245:
A76   A93   L5ALALLTFX  CT:                               :V43301:
START A82   AFALALPLSC  CT:  863/ 750:  486/  492:NE4JBD:V43120:
A82   A83   CFALALPLSC  CT:                               :V43133:
A83   A84   I2ALALPLSC  CT:                               :V43245:
A84   A93   LO1LALPLSC  CT:                               :V43301:
START A78   AFALALPL3X  CT:1107/1000:  521/  531:W28KNR:V43122:
A78   A79   CFALALPL3X  CT:                               :V43135:
A79   A80   IOALALPL3X  CT:                               :V43247:
A80   A93   LOALALPL3X  CT:                               :V43301:
A63   A64   IOALALTCDC  CT:  412/ 402:  262/  262:NE3ROM:V43248:
START A90   AFALALTFPC  CT:  203/ 200:  169/  151:NE4JBD:V43120:
A90   A91   CFALALTFPC  CT:                               :V43133:
A91   A92   I2ALALTFPC  CT:                               :V43245:
A92   A93   LOALALTFPC  CT:                               :V43301:
A93   A94   S-PLTI SYS TEST                               :V45001:
A94   A999  T-PLTI XMT CUST                               :V61000:
```

PSR 8B Software Network (Sorted by Task Name)

DAILY BARCHART

```
PRINTED  15SEP76 AT 13.30.34    PROJECT:   ASP ADM; UNIT: APPLICATION SOFTW LV03; SUB UNIT: PLTI (PLTILV03)
TIMENOW DATE USED: 06AUG76          LAST UPDATE WAS: 13SEP76
SORTED BY DESCRIPTION SELECTION

o-SPAN OF EARLY START TO EARLY FINISH
*-SPAN OF LATE START TO LATE FINISH (SPREAD TO SHOW + OR - SLACK)
●-OVERLAPPING EARLY AND LATE DATES
□-COMPLETED ACTIVITY EARLY START TO EARLY FINISH
+-PRESENT TIMENOW LINE

--------------------------------------------------------------------------------------------------------------
PRED  SUCC     ACTIVITY DESCRIPTION - SOFTWARE                   28JUN76    12JUL76    26JUL76    09AUG76    23AUG76

              |TSAP| TASK |   | NUMBER    | PROGRAM  | D  P |       |
              |ATRU| NAME |   | SOURCE    |  STORE   | E  G |       |
              |SIEN|      |   | STMNTS    |  (DPW)   | P  M | WORK  |
              |KTAC********LV|CURR/PLAN|CURR/PLAN| T  R | PKGE  |
--------------------------------------------------------------------------------------------------------------
START A62   AFALALBBC1   CT:   35/   27:    9/    8:NE4JBD:V43120:
START A66   AFALALBC1X   CT:  492/  500:  260/  256:NE4JBD:V43120:
START A74   AFALALLTFX   CT:  906/  893:  613/  225:NE4JBD:V43120:
START A82   AFALALPLSC   CT:  863/  750:  486/  492:NE4JBD:V43120:
START A90   AFALALTFPC   CT:  203/  200:  169/  151:NE4JBD:V43120:
START A78   AFALALPL3X   CT: 1107/ 1000:  521/  531:W28KNR:V43122:
A62   A63   CFALALBBC1   CT:                                 :V43133:
A66   A67   CFALALBC1X   CT:                                 :V43133:
A74   A75   CFALALLTFX   CT:                                 :V43133:
A82   A83   CFALALPLSC   CT:                                 :V43133:
A90   A91   CFALALTFPC   CT:                                 :V43133:
A78   A79   CFALALPL3X   CT:                                 :V43135:
A63   A64   IFALALCCBC   CT:  629/  629:  436/  436:NE4JBD:V43245:
A75   A76   IFALALLTFX   CT:                                 :V43245:
A83   A84   I2ALALPLSC   CT:                                 :V43245:
A91   A92   I2ALALTFPC   CT:                                 :V43245:
A63   A64   I6ALALBBC1   CT:                                 :V43245:
A67   A68   IOALALBC1X   CT:                                 :V43245:
A79   A80   IOALALPL3X   CT:                                 :V43247:
A63   A64   IOALALCCBC   CT:  412/  402:  262/  262:NE3ROM:V43248:
A64   A93   LOALALBBC1   CT:                                 :V43301:
A68   A93   LOALALBC1X   CT:                                 :V43301:
A80   A93   LOALALPL3X   CT:                                 :V43301:
A84   A93   LOALALPLSC   CT:                                 :V43301:
A92   A93   LOALALTFPC   CT:                                 :V43301:
A76   A93   LSALALLTFX   CT:                                 :V43301:
A93   A94   S-PLTI SYS TEST                                  :V45001:
A94   A999  T-PLTI INT CUST                                  :V61000:
```

PSR 8C Software Network (Sorted by Work Package)

PSR 9 DEPENDENCIES

PROJECT NAME ______________ REPORT DATE ______________

	Description	Status
External To Location		
Internal To Location		

Indicated here is a module task-phase whose calendar allotment is approximately 80% and whose indicated "Status" is 80%, so we are on schedule.

16. **Example 4**
Indicated here is a module task-phase whose calendar allotment is 40% and whose status is 50%, so we are ahead of schedule.
17. **Sort Grouping.** Module-task-phases are grouped (sorted) by task-phase. Alternate reports are available by sorting on any characters within the 48-character Description Field (by work package, by task name, by department and programmer, etc.)
18. **Data Totals.** A planned future feature of LUBAR will be the summarization of source statements and Program Store words.

The PSR 8C report is generated using barchart generator program LUBAR.

PSR 8B and PSR 8C show reports containing the same network data sorted by task name (5) and work package (11).

A complementary documentation report is also generated using LUBAR by specifying the documentation option.

LUBAR modification details are available as an IBM Technical Report "The use of APL MINIPERT in a Project Development Plan"[6] from the IBM Library, IBM Corporation, Manassas, VA.

PSR 9 EXTERNAL DEPENDENCIES

All external dependencies to the plan, that is dependencies deriving from nonproject staff, must be explicitly stated. These dependencies include:

Data Processing Support
Terminal Facilities
Turn Around Time, etc.
Physical Work Space
Work Space Access Over Various Shifts
Access to Real Hardware
Development Level of Real Hardware Access

ASSUMED EXTERNAL RESOURCES

Project Name — Common name of project

Project Date — Date of status review

External To Location — off site

Internal To Location — on site

Description — Brief specific summary of service or facility needed or expected.

Status — availability of described service or facility.

A GUIDE TO CONDUCT PROJECT STATUS REVIEWS

Project Status Reviews require different levels of information for various management-level reviews. The following inclusion guide suggests an appropriate information and detail content for such reviews:

Figure 1 shows how "layered" detail is presented to corresponding management levels. For example, third level managers should normally see PSR 1 through PSR 6, while second level managers are also aware of PSR 7 and PSR 8, 9 as required.

It was found that when too much detail (PSR 9, 8, 7, etc.) was included in third level and above management reviews, these managers tended to concentrate of the "fun" of direct involvement and sometimes dropped their proper "evaluation and strategy" role in favor of a "let-me-fit-it" role.

Conversely, when technical personnel were too closely involved with financial and business strategy, they tended to become quickly enamored with the "big picture" and technical productivity suffered.

The fundamental hierarchy of detail embodied in the above system allows any reporting person to respond to any schedule question via an immediate reference to coordinated and available detail.

The development plan is usually reviewed through several levels of management and approval thereby is usually considered authorization to proceed with the project.

CONCLUSION

The principles, procedures, and forms herein described have been variously used on several projects under the author's direction or guidance and represent a synopsis of "all the things we ought to do — or should have done" — to run complex engineering and software development projects.

However, the use of all planning, simulation, and control and reporting tools will always be tempered by the dimension of the project and the style of particular management personnel. The foregoing guides and report formats should serve as a suggestion to apply those tools in specific circumstances.□

ACKNOWLEDGMENT

The author is indebted to J. Luke, IBM FSD, Manassas, Virginia, for modifying APL MINIPERT to generate more-meaningful barchart reports, to R. Williams, IBM FSD, Gaithersburg, Maryland, for his suggestions and contributions to managerial review organization, and to F. Mutz and N. Woodrick, IBM FSD, Gaithersburg, Maryland, whose guidance and foresight made it possible to formally document the principles herein explained.

REFERENCES

[1]Project Management with CPM & PERT, J.J. Moder and C.R. Philips, New York, Rheinhold Publications, 1964.
[2]Progress Evaluation & Review Technique (PERT), J.D. Weist and F.K. Levy, Prentice Hall, Englewood Cliffs, NJ.
[3]MINIPERT Program Description & Operators Manual, SH20 0995-0, IBM Corporation, Armonk, NY.
[4]Mark III Project Management System, Program Control Corporation, Van Nuys, CA.
[5]Work Plan Analysis & Scheduling Technique (WOPAST), A. Del Grusso and W. Rosenbluth, IEEE Transactions on Engineering Management, November, 1976.
[6]The Use of APL MINIPERT in a Project Development Plan, J. Luke and W. Rosenbluth, September, 1976, IBM Corporation, Manassas, VA.

ABOUT THE AUTHOR

William Rosenbluth is at IBM Federal Systems Division, Gaithersburg, Maryland where he directs applications of microcircuit technology to military space projects.

Mr. Rosenbluth has published several technical papers, as well as an article on WOPAST in the 11/76 edition of the IEEE Transactions on Engineering Management. Successor articles on Project Reporting, Risk Management and Financial Planning are in preparation.

He is chairman of the annual IEEE Computer Elements December Workshop in Phoenix, Arizona. He received his BSEE in 1961, and MSEE in 1965 from the Polytechnic Institute of Brooklyn.

NEW PRODUCTS

DOS for Poly 88/North Star Disk Systems

Cardinal Products announces The Lazy Man's DOS for Poly 88 owners with North Star disk systems. This expansion of the North Star disk operating system brings about a successful marriage of the two systems and allows full control with a minimum of keystrokes.

Control-character commands let you quickly and easily load and start BASIC, jump back to DOS, cold or warm restart BASIC, list directory while in DOS *or BASIC* and bring up front panel mode at any time.

List scrolling can be controlled on a line by line basis. Control-Z is released for use in the BASIC editor. Delete key backspaces and erases a character at a time. The Poly 88 real time clock interrupt system may be left connected.

Diskette and instructions are only $15.95 from Cardinal Products, 1600 Tilden St., Wichita Falls, TX 76309.

CIRCLE INQUIRY NO. 110

Pocket-Size Experimentor™ Boards

Experimentor Socket solderless breadboards are made by Continental Specialties Corporation. The smallest of these, just 3.6" x 2.1" x .3" is about the size of an audio cassette.

No soldering, drilling or tooling is required. Parts simply plug right into the breadboard and interconnections are accomplished by pushing short lengths of hookup wire into adjacent holes.

The CSC Experimentor Socket referred to here, Model EXP-350, is priced at $5.50. Other Experimentor boards are priced from $4.00 to $10.95. For more information contact Continental Specialties Corp., 70 Fulton Terrace, New Haven, CT 06509, (203) 624-3103.

CIRCLE INQUIRY NO. 111

24-Channel Isolated Digital Input System

New single board microperipherals accept 24 digital inputs. MP810, with an on-board power supply, operated with dry relay contacts. MP810-NS, with voltage inputs, operates with wet relay contacts.

MP810's are electrically and mechanically compatible with Intel SBC-80, Intellec MDS and National BLC-80 microcomputers and operate from their +5 VDC supplies.

In 1-9 quantities MP810 is priced at $355; MP180-NS at $295. Delivery is from stock. For additional information contact Burr-Brown, International Airport Industrial Park, Tucson, AZ 85734, (602) 294-1431, C.R. Teeple, Product Manager.

CIRCLE INQUIRY NO. 112

"Total" Data Query System

A new interactive language which enables non-EDP personnel to retrieve information easily from highly complex files is called NCR Total IQL (Interactive Query Language). It is used in conjunction with the NCR Total data base management system. NCR Total helps manage large numbers of interrelated files of information so that to the user it appears as if there is a single, well-organized master file.

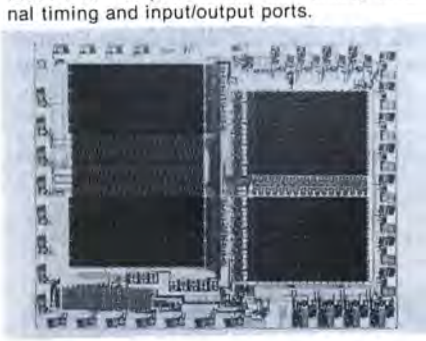
The basics of NCR Total IQL can be learned in an hour. The language allows users to make virtually any type of inquiry to the NCR Total data base at any time. However, access can be limited through the use of passwords or access codes.

Total and Total IQL can be used with NCR Century or Criterion computer systems, beginning with a 96K Century 101. Total IQL is sold for a one-time payment of $15,000 or a monthly license fee of $572. For more information contact NCR Corp., Dayton, OH 45479, (513) 449-2150.

CIRCLE INQUIRY NO. 113

System Memory Interface forms Powerful Two-Chip Microcomputers

The 2656 System Memory Interface (SMI) is a single monolithic IC for microprocessor interfaces that incorporates its own memory, internal timing and input/output ports.



The new IC can be combined with Signetics 2650 microprocessor to form a powerful two-chip microcomputer that is unmatched for system flexibility and both memory and input/output expansion capabilities. The new SMI can also be used with the microprocessors of other manufacturers.

The Signetics 2656 System Memory Interface is priced at about $17.00 in quantities of 1000. For further information contact Signetics, P.O. Box 9052, 811 E. Arques Ave., Sunnyvale, CA 94086; (408) 739-7700.

CIRCLE INQUIRY NO. 114

Digital Announces Software for Retrieval of Commercial Data

A new software package enabling personnel who are not skilled programmers to retrieve data stored in DEC DATASYSTEM Business computer systems is called DATATRIEVE-11. The program package permits users to access data stored in computer files in sequential, indexed, or relative organization, and extract such data via simple English-like commands. It will generate reports from the accessed data via simple commands.

DATATRIEVE-11 is compatible with the files accessed by PDP-11, COBOL, BASIC-Plus-2, or other programming languages by utilizing the RMS-11 files created by these compilers. The software package is intended for use primarily with stand-alone business systems, or for systems distributed throughout large companies.

DATATRIEVE-11 is licensed at $4,500, which includes the RMS-11K Record Management Services package, and is available immediately. For more information contact Digital Equipment Corp., Maynard, MA 01754, (603) 884-5101, Joseph Nahil.

CIRCLE INQUIRY NO. 115

Microbench™ Software

Microbench software is a family of computer programs for microprocessor application program development. These programs operate in conjunction with PDP-11 and LSI-11 computers to provide an economical program development capability for popular microprocessors.

Featured in Microbench software are relocating assemblers and linking loaders for the Intel 8080/8085, Zilog Z-80, Motorola 6800 and equivalent microprocessors. Coded in Macro-11 for high throughput, these assemblers and loaders operate on PDP-11 and LSI-11 computers under the RT-11 operating system in 16K words of memory.

For additional details and pricing information contact Virtual Systems, Inc., 1500 Newell Ave., #406, Walnut Creek, CA 94596, (415) 935-4944.

CIRCLE INQUIRY NO. 115

Mag Card Unit Priced for Consumer

The Series LC-31 GAMELOADER™ Magnetic Card Reader/Encoder is designed for convenient, low-cost program loading in consumer electronics applications including games, integrated home entertainment centers, and home/hobby computers.

The LC-31 provides "jitter free" recording and reading, with character load time of 5 kilobytes per minute. A totally enclosed strobe controlled PC/LED/shaft encoder provides data accuracy independent of head speed, by measuring the rotational displacement of the lead screw to precisely determine magnetic head lateral displacement.

The LC-31 is priced at under $40 in quantities over 100,000. For more information contact August A. Toda, President, Vertel, Inc., 125 Ellsworth St., Clifton, NJ 07012, (201) 472-1331.

CIRCLE INQUIRY NO. 117

High Speed 4-Bit-Slide 2900 Bipolar Microprocessor

Using an advanced bipolar LSI process, National Semiconductor has developed a family of 2900-type 4-bit-slice microprocessor components which are 30 to 50 percent faster than any similar designs now on the market.

Designated the IDM2900 family, the new devices, 16 in all, use a process that combines low power Schottky peripheral circuitry with proprietary high speed tri-state emitter coupled logic circuitry for interface.

The IDM2900 family includes a 4-bit bipolar microprocessor slice, a high speed look ahead carry generator, 4-bit wide address controllers, inverting 64-bit RAMs, and others.

For more information contact National Semiconductor Corp., 2900 Semiconductor Dr., Santa Clara, CA 95051, (408) 737-5000.

CIRCLE INQUIRY NO. 118

Wire Wrap Panel and Card Catalog

This 28-page brochure covers EECO's newest pin-in-board type ALA wire wrap panels, cards, drawers and frames.

Products are thoroughly documented with photos, descriptions, specifications, outline drawings and prices.

ALA hardware is distributed by Marshall Industries. For more information contact EECO, 1441 E. Chestnut Ave., Santa Ana, CA 92701, or phone "EPP Products" (714) 835-6000.

CIRCLE INQUIRY NO. 119

Programmable Industrial Control Unit Doubles as Learning Aid

Motorola has a 2-board Industrial Control System that combines a highly functional, prewired programmable logic controller (PLC) with an ancillary Input/Output Simulator that serves as a system development tool and demonstration unit.

The MC14500B Industrial Control Unit (ICU) can serve as a learning tool to acquaint designers with the power and potential of a one-bit MPU and, thereafter, as a dedicated functional control system. As a functional system, the I/O Simulator is replaced by the actual I/O devices associated with the working system.

For more information contact Motorola Semiconductor Products Inc., P.O. Box 20912, Phoenix, AZ 85036, (602) 244-6900.

CIRCLE INQUIRY NO. 120

New Quad Power Supply Mounts on P.C. Board

Scientific Programming Inc. introduces a new quad output P.C. board-mountable Power Supply, the newest unit in their Micro-Supply (MS) family. A typical quad output supply offers 5V, 12V, -12V, and -5V.

The Micro-Supply series provides up to four outputs and have an overall height of less than 0.5 inch (standard P.C. board spacing).

The QUADOUT 5/12/-12/-5 is priced at $59.95 in OEM quantities. Prices of other modules range from $29 to $79 depending on the type and quantity. Delivery is from stock to 4 weeks. For more information contact Scientific Programming Inc., 1499 Bayshore Hwy, Suite 126, Burlingame, CA 94010, (415) 493-2199.

CIRCLE INQUIRY NO. 121

200 Watt and 400 Watt Open-Frame Switching Regulated Power Supplies

Boschert Incorporated has 200 and 400 watt open-frame multiple output switching power supplies. Both switches are compact, have negligible audible noise, high efficiency and are available at prices of less than a dollar a watt in quantities of a hundred.

The standard OL-200 is a four output unit which supplies 200 watts of continuous power, + 5V at 25A (max.) from one output, -5V at 4A, and ± 12 at 4A each from the other outputs.

The standard OL-400 is a five-output unit that is capable of supplying a continuous 400 watts, + 5V at 45A from one output, ± 12V at 10A each from two outputs, and -5V and + 24V at 4A each from the remaining two outputs.

The OL-200 is priced at $248 and the OL-400 at $395 in quantities of a hundred. For more information contact Boschert Associates, 384 Santa Trinita, Sunnyvale, CA 94086, (408) 732-2440, Scott Warner.

CIRCLE INQUIRY NO. 122

Short Length Cassettes for Microcomputers

Cassette Date Tapes are available to store microcomptuer programs for home and hobby computers. Microsette Data Tapes are digital quality cassette tapes customized for the microcomputer user and available in lengths of 50, 100, 200 and 300 feet. The high energy tape used in our Data Tapes has been selected for consistency in output envelope and is error-free for the recording densities used by all popular home and hobby computers. The products are backed by a user oriented warranty covering defects in materials or workmanship.

There are four Microsette Data Tapes of one length per package. Each cassette comes with a hard box (Norelco style) and two extra sets of labels. Prices for each 50, 100, 200 and 300 foot cassette are $0.60, $0.70, $0.85, and $1.00 respectively. For more information contact Microsette Co., 777 Palomar Ave., Sunnyvale, CA 94086.

CIRCLE INQUIRY NO. 123

MK 8600 Memory System

The MK 8600 is a versatile memory chassis with a total capacity of five megabytes — ideal for mass storage applications such as mainframe add-on memory and disk replacement. The general purpose 12¼-inch chassis with power uses Mostek's MK 8000 memory card.

This versatile board features from 16K x 18 to 128K x 24 words of storage for small to memory intense applications. Standard access time is a fast 250ns with a cycle time of just 450ns. The versatile configuration of the chassis allows for up to 16 MK 8000 boards, pluc ECC, and there are four additional slots for I/O. For smaller requirements, Mostek offers a 7-inch chassis with a one megabtye capacity, the MK 8601.

Delivery on the MK 8600 is in 60 days. Price varies with memory and interface requirements. For more information contact Mostek Memory Systems, 1215 W. Crosby Rd., Carrollton, TX 75006, (214) 242-0444.

CIRCLE INQUIRY NO. 124

Genesis One Now Marketing Wordstream™ on West Coast

Genesis One Computer Corporation introduces its Wordstream word processing system to the West Coast markeplace.

Wordstream utilizes the latest techniques in computer technology to simplify and improve word processing operations. The result is a system that is easy to use, more efficient, and highly productive.

The Wordstream system features a console containing the electronics and associated controls, and a full-page display which shows the oeprator exactly the way a finished, printed page will look on a standard sheet of paper. Easy-to-handle diskettes are used to store data. Each diskette holds more than 350,000 characters.

Wordstream is a multi-terminal system for clustered, remote or centralized operation, with all operators having simultaneous access to all stored data. Wordstream also features independent print station operation. This enables operators to work on new projects, while previously edited text is being printed. Standard printers, or special wide track printers can be used to produce tabular documents up to 26 inches wide.

For more information contact Genesis One Computer Corp., 300 E. 44th St., New York, NY 10017.

CIRCLE INQUIRY NO. 125

KSAM80

KSAM is a file management system designed specifically for floppy disk microcomputer systems.

It was developed primarily for use in applications where large files are involved and fast random access is a necessity. Such applications include, but are not limited to, inventory control, reservations systems, library systems, accounts receivable, and bill of materials processing.

Random storage and retrieval of records is based on the contents of a user-defined data field within the record which is called the key.

The key must be unique for each record and it can be any string up to 255 characters long. Examples of keys are: part numbers for inventory control, account numbers for billing systems and customer names for mailing list applications.

KSAM80 also supports sequential access of records starting at any point within a file, random access by partial key and random access by relative record number. Sequential and random access commands can be intermixed freely.

Space is automatically allocated to the file when records are added, and reclaimed when records are deleted, so that KSAM80 files are self-reorganizing, and any number of files can be processed simultaneously provided that sufficient buffer storage is available.

KSAM80 was originally developed under Zilog-s Z80 OS 2.0, but can be easily implemented in many existing microcomputer operating systems. For additional information or personal demonstration contact EMS Co., 3645 Grand Ave., Suite 304, Oakland, CA 94610, (415) 834-4944.

CIRCLE INQUIRY NO. 126

Offsco Vinyl Decals Feature Stand-Out Numbers and Letters

Offsco decal numbers and letters are available in convenient strips of 0 through 9 and A through Z or in the buyer's choice of any number or letter of identical size in strips. Offsco decals come in height sizes of 2½, 3, 4, 5 or 6 inches. The company also offers the numbers and letters individually cut and packaged.

Offsco decals fill the needs of large and small vehicle owners for low price quality identification in quantities to suit all requirements.

The decals are produced from commercial-grade, heavy duty weatherproof vinyl in dull or high gloss finish, and come in all colors on red or white background and in colors on a transparent background.

For more information contact Office Specialties Co., P.O. Box 66492, Houston, TX 77098, (713) 524-8980.

CIRCLE INQUIRY NO. 127

Xycon III

The Xycon III microcomputer delivers state-of-the-art computer technology to achieve "big system" features at a moderate price.

Each unit undergoes stringent testing and "burn-in" before it leaves the factory. Features like the well-proven *Intellect® Bus, a power supply in a drawer, that slides out for quick service, and its advanced modular construction give the Xycon III a 20-minute mean-time-to-repair. Each unit carries a full six-month limited warranty.

The all-in-one Xycon III standard system consists of the following: A 24x80 high resolution

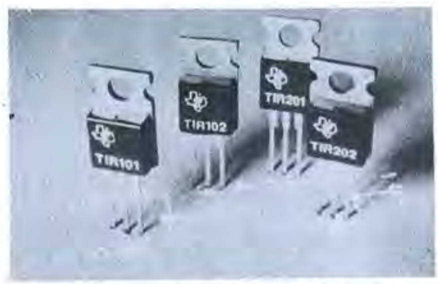
CRT and controller with character intensification, blinking, underscoring, and reverse video — in any combination, all on a character-by-character basis. A 63-key typewriter-style keyboard with 16-key numeric and cursor cluster and 8-key alternate action pad. A memory board filled with 32K of RAM, expandable in 16K increments to 65,536 bytes and usable as either 8 or 16-bit word memory. Dual floppy disks with an advanced intelligent controller that uses its own firmware to do formatting and much more. It can control two dual-sided disks to give a full megabyte of storage. A fast *8085A® processor board with firmware operating system and space for extra 6K EPROM, TTY port, 8 levels interrupt (expandable to 64), real time clock, and special logic for an 8-channel bus controller, allowing true distributed processing on a Master/Slave basis.

For more information contact Computer Systems Unlimited, Marketing Div., P.O. Box 870, Milpitas, CA 95035, (408) 262-6271.

CIRCLE INQUIRY NO. 128

Battery Charger Applications Report

Bulletin CA194, available from Texas Instruments Inc., explains how the TL430 programmable shunt regulator and the TIR101 dual common cathode rectifier are used for designing an economical battery charger.



The nine-page report is titled "Current Limited and Voltage Regulated Battery Charger." It provides details on how a circuit is designed to properly rejuvenate a 44 ampere-hour lead-acid battery from fully discharged to fully charged in three hours. Charts, schematics, photos and mathematical data are included.

Bulletin CA194 is available from Texas Instruments, Inc., Box 5012, M/S 308, Dallas, TX 75222.

CIRCLE INQUIRY NO. 129

New Tone Dialer from Mostek

Mostek has a new integrated tone dialer for 2500-type telephone applications. Designated MK 5090, the tone dialer uses an inexpensive crystal reference to provide eight different audio sinusoidal frequencies, which are mixed to provide tones suitable for Dual Tone Multi Frequency (DTMF) telephone dialing.

The MK 5090 was designed specifically for integrated tone dialer applications that require the following: variable supply operation with loop compensated tone regulation, single-contact-keyboard, chip disable input, and a mute output that is open circuit when no keyboard buttons are pushed and pulls to the V+ supply when a keybaord button is pushed.

Priced at $4.95 in 100-piece quantities, the MK 5090 is immediately available. For more information contact Mostek Corp., 1215 W. Crosby Rd., Carrollton, TX 75006, (214) 242-0444, Don Ward.

CIRCLE INQUIRY NO. 131

Versatile Impact Printer

The Integral IP-125 Impact Printer features an RS232C serial interface, parallel TTL level interface and full upper and lower case ASCII character set (96 characters) as standard equipment. Capable of printing multiple copies on ordinary 8½" roll, fanfold or sheet paper, the microprocessor controlled IP-125 incorporates a 256-character multi-line buffer to achieve an instantaneous print rate up to 100 characters per second with a sustained throughput of 50 cps at 80 columns per line.

The Integral IP-125 has few moving parts and features a reinking ribbon. Line length is 80 columns at 10 characters per inch (7x7 dot matrix format).

The Integral IP-125 sells for $799. For more information contact Integral Data Systems, Inc., N. Lamade, Director of Sales, 5 Bridge St., Watertown, MA 02172, (617) 926-1011.

CIRCLE INQUIRY NO. 132

Pipe Analysis Program

United Computing Systems, Inc., has added the latest version of TRIFLEX, a comprehensive computer program for performing piping flexibility and stress analyses, to its program library.

Developed by AAA Technology and Specialties Co., Inc., TRIFLEX is designed for use in

"Gordon got the idea from those new computerized stadium scoreboards!"

the petrochemical, pipeline, shipbuilding, power and commercial building industries.

TRIFLEX uses the stiffness matrix technique to perform analysis for all types of piping systems. Once basic piping information is input, TRIFLEX simulates the actions of the piping system when subjected to specific loading conditions.

A major feature of the program is that it provides printed reports for checking piping design compliance with established piping codes and standards. The new version is easier to use and permits users to select any of a number of printed reports, including flange loading, automated spring hanger/support selection, geometry check plots and ANSI B31 code compliance.

For more information contact Ron Kogan, product manager, United Computing Systems, 2525 Washington, Kansas City, MO 64108.

CIRCLE INQUIRY NO. 130

GMXBUG Monitor

The GMXBUG Monitor will be available shortly. It is designed for use with our Video Board and no terminal is required — only a keyboard. It will be on two 2708 PROMs with a manual. It is used in place of MIKBUG or SWTBUG.

Dealer costs are as follows: 2-chip set and manual, \$48.00; Video Board & GMXBUG, \$217.00; 8K PROM Board, Video Board and GMXBUG, \$294.00.

The new cabinet, power supply, motherboard and CPU make up the most powerful and flexible 6800 mainframe on the market. For more information contact Gimix, Inc., 1337 W. 37th Pl., Chicago, IL 60609, (312) 927-5510.

CIRCLE INQUIRY NO. 133

Asynchronous Line Driver

A "Zero Downtime" asynchronous line driver is comprised of a single unit housing two asynchronous line drivers. While one ALD is in operation, the other remains in a ready state as an on-site back-up unit. If the primary unit should fail, the redundant ALD is present to pick up the load.

The "Zero Downtime" option conforms to Bell specification 43401. Specific features of the limited distance line driver include 0-9600 bps asynchronous transmission, 2-wire half duplex or 4-wire full duplex, point-to-point or multidrop, easy-to-operate controls and LED performance indicators.

For more information contact Ven-Tel, Inc., 2360 Walsh Ave., Santa Clara, CA 95050.

CIRCLE INQUIRY NO. 134

The Superslice™

The Am2903 has two's complement multiply and divide capability. It eliminates the need for additional hardware for both signed and unsigned multiplication and performs signed division using a non-restoring algorithm.

Additionally, the Am2903 can perform normalization on both single- and double-length words and can automatically convert between sign-magnitude and two-s complement notation. This device can increment by either "one" or "two" on a single cycle and has internal parity generation.

This new low-power Schottky circuit is a 4-bit microprogrammable data processor slice containing a multi-function arithmetic logic unit, a two-port 16-word scratch-pad memory, an additional accumulator register and shifting and control logic.

In addition to logical shifts, the Am2903 offers arithmetic shifts; also, its two data-input ports can operate between any two internal registers, any internal and external data bus, or between two external data buses. The Am2903's ALU offers an expandable register file.

This microprocessor slice is available in a 48-pin ceramic dual-in-line package for use over the military and commercial temperature ranges, and undergoes 100 percent processing to the requirements of MIL-STD-883. Prices for the Am2903 start at \$29.95 in 100-piece lots. For more information contact Advanced Micro Devices, Inc., 901 Thompson Pl., Sunnyvale, CA 94086, (408) 732-2400, E. Sopkin.

CIRCLE INQUIRY NO. 135

I/O Processing Unit

The APU100 is a general-purpose input/output processing board that provides a high-performance interface to the standard S-100 bus.

Designated the Extensys Asynchronous Processing Unit, the APU100 includes an on-board 8080 processor. The APU100 operates asynchronous with the central processing unit of the computer system and transfers information in full Direct Memory Access (DMA).

The APU100 uses the system clock on the bus to provide internal timing so that all system processors are synchronized. The unit has 8192 bytes of dynamic RAM storage operating at 300ns access time and 1024 bytes of 2708-type EPROM storage in addition to its dedicated 8080 processor. The EPROM storage is used for device initialization routines, while the RAM storage is used for file management I/O programs and buffering.

Using the APU100 frees up 8K bytes of system memory by moving I/O routines to the APU100, allowing more memory for application programs.

The APU100 gives a simply defined structure for system-integration applications such as realtime processing and it puts the power of an 8080 to service the realtime devices without tying up the complete system.

The APU100 is available in volume OEM quantities. Delivery is 30 days ARO. For more information contact Extensys Corp., 380 Bernardo Ave., Mountain View, CA 94040, (415) 969-6100, Ed Hartnett.

CIRCLE INQUIRY NO. 136

Software

Now available, a full complement of Z80/8080/8085 software development tools. These are disk based, oriented towards the popular CPM Operating System™. System development tools include a relocatable linking macro assembler with linking loader, cross-reference generator, and full library of modules. With the assembler is included a symbolic debugger allowing user defined symbols. Higher level language support is provided by interface with Micro-Soft FORTRAN.

For advanced systems, TSA/OS is an upward compatible CPM-like operating system providing CRT screen control, automatic library search, an extended batch mode with turnkey system capability, as well as an advanced configuration scheme.

TSA Software has available a set of applications packages. The TSA Database System uses a mixture of assembly code and FORTRAN to produce a highly effective system. The system uses table driven screen and record formats; and has a mini-compiler to optimize record search capability. The TSA Word Processor uses a normal terminal to provide natural text editing with advanced formatting features, including proportional printing. Full use of disk files is provided, as well as file merging for mailing list and similar uses.

Package prices start at \$100. Dealer and OEM inquiries welcome. For more information contact TSA Software, 5 N. Salem Rd., Ridgefield, CT 06877, (203) 438-3954.

CIRCLE INQUIRY NO. 137

Mailing List Program

The Comprehensive Mailing List Program #ML-1NS, is a modular program set which enables the user to start and effectively maintain one or more mailing lists. Operations include:

Sunshine Computer Company

PROUDLY OFFERS **SYSTEM THREE**

From Cromemco

Brand New! FOR TIME SHARING

A Powerful Microcomputer System with Capacity for 4 DISK DRIVES!

Price starts at $5990* assembled and tested.

SYSTEM THREE includes:

- Z-80A Microprocessor
- 32K of RAM on 2 cards
- Dual Disk Drive
- 4-Drive Controller
- RS-232 Interface
- S-100 Bus
- Power-On Switch
- 21-slot Motherboard
- Heavy Duty 30 amp Power Supply
- 110 or 220 Volt Operation

Model 3100 CRT Terminal with 80 character/line, upper/lower case and separate numerical and cursor keypads. Price $1595.

Model 3101 with added line editing, block mode and function keys. Price $1995.

High Speed Line Printer Model 3703 speed is 180 cps, bi-directional with 132 column width and tractor feed. Price $2995.

MANY OPTIONS AVAILABLE

We'll help you design your computer system to suit your special applications for:

- Business/Accounting
- Word Processing
- Data-Based Management
- Science and Engineering
- Legal or Medical
- Education

EXTENSIVE SOFTWARE SUPPORT

All current software developed by Cromemco works for The SYSTEM THREE Computer, including:

- FORTRAN IV
- 16K Z-80 BASIC
- Z-80 Macro Assembler and Linking loader

A Full Accounting Software Package is now available for SYSTEM THREE.

COMPLETE SYSTEM

With your choice of peripherals, whether they're from Cromemco or from any of the fine devices we have in stock, you can have a microcomputer system up and running, performing a wide range of tasks, for less than $10,000.

*Special pricing available with full system purchase. **Sunshine Computer Company** specializes in assembled systems. All systems are sold with a 90-day written warranty. Stop in for a demonstration. If you're not nearby, call or write for more information, because **Sunshine Computer Company** ships anywhere in the U.S.

Sunshine Computer Company

20710 South Leapwood Ave. • Carson, California 90746 • (213) 327-2118

Sunshine Computer Company

Features

North Star *and* Hazeltine 1500 *with a Powerful*

HORIZON-2

This is a fully integrated dual drive micro-disk system which uses the fastest processor available today, and offers expansion capabilities with the addition of many S-100 bus board options.

Here's how we equip your HORIZON-2 for immediate plug-in operation:

- Z80A processor operates at 4 MHz.
- S-100 bus 12-slot motherboard
- 16K RAM memory at 4 MHz includes error checking and bank switching.
- Dual integrated mini-floppy disk drives.
- Power supply.

To immediately increase your computing capability, we add 2 serial I/O ports: one for your Hazeltine 1500, the other for a printer, or any other peripheral you choose.

Completely assembled and tested. Just plug it in and compute! Full price for the complete system described here, assembled with 90-day written warranty is $3574.

Hazeltine 1500

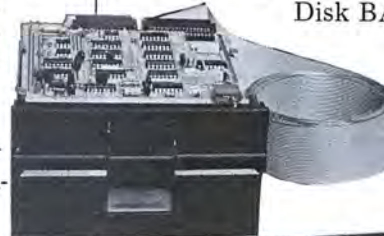
We add this fine terminal to your HORIZON-2 to give you a complete operating system.

You have a separate numeric keypad and switch-selectable upper and lower case capability—all added to these great features:

- 24x80 display on 12" screen.
- 7x10 dot matrix.
- 8 switch-selectable rates to 19,200 baud.
- Horizontal tabs.
- Brightness control.
- Clear to "End of Line" and "End of Screen".
- Remote cursor stepping.
- Read cursor address.
- Remote keyboard lock.
- RS232 compatible.
- Automatic repeat-all keys.
- Audible alarm.
- Aux. serial output port.

Disk BASIC

You get your HORIZON-2 and Hazeltine 1500 complete with North Star Disk BASIC.



ADDITIONAL SOFTWARE

Business Package

General Ledger
Accounts Payable
Accounts Receivable
Payroll
Inventory
Amortization
Mailing List

The entire package is yours with the purchase of the HORIZON-2 System for only $295.

SYSTEM Z Development Software

ASMB Z80 Assembler and Editor - $60
DASMZ Assembler $60
ZEDIT Text Editor $60
ZEBUG Debug Monitor/ Disassembler $60

Drop in and see our own HORIZON-2 and Hazeltine 1500 in operation.

Sunshine Computer Company

20710 South Leapwood Ave. • Carson, California 90746 • (213) 327-2118

Add, Delete, SEarch, Sort, Auto-Sort, and Sequential Printout.

Features include: user-selectable defaults for ease of entry, user-selectable number of labels across page for different printers and label sheets, and user-selectable 3 or 4 line address for each independent entry.

The program set is written for convenience and ease of use. Available with complete documentation and North Star diskette for $25 PPD. Delivery is from stock. Documentation package only is $4.50 PPD, fully refundable with order for diskette. Order from: Williams Radio and TV, Inc., Computer Div., 2062 Liberty St., P.O. Box 3314, Jacksonville, FL 32206.

CIRCLE INQUIRY NO. 138

CP/M™ for iCOM Frugal Floppy

The iCOM/CPM System Upgrade gives the iCOM disk owner the full capabilities of the CP/M disk operating system, while retaining full access to iCOM's FDOS operating system.

CP/M is rapidly becoming the standard operating system for 8080 and Z80 disk software. In recent months, many excellent disk software products have become available. All of this great software has one thing in common: You must have CP/M to run it.

The iCOM/CPM System Upgrade allows the iCOM disk owner to participate in this standard, by acquiring a superior disk operating system with the following features: Directly load and start CP/M, FDOS-II, or FDOS-III; transfer files between CP/M and FDOS diskettes; automatically execute the program of your choice at system start-up; dynamic disk space allocation and reclamation; random access on all files; high speed disk read and write; full compatibility with all other CP/M systems and software.

Of course, you automatically get all the advantages of standard CP/M: Intel-compatible assembler; powerful interactive debugger with built-in assembler/disassembler; object programs stored in binary for fast execution; powerful batch capability, including parameter substitution at execution time; unlimited number of files open simultaneously; and many more.

For more information contact Computer Mart of New Jersey, Inc., 501 Route 27, Iselin, NJ 08830, (201) 283-0600.

CIRCLE INQUIRY NO. 139

Add-on Memory for IBM 370/135 and 145

Designated the CalComp 4135/4145, the new memory system incorporates a unique interface that allows complete hardware and software compatibility with any 135 and 145 processors. This interface module can be field upgraded, permitting the system to adapt easily to 138 and 148 processors.

An important feature of the new 4135/4145 is the use of a 4K static RAM, which provides twice the density, with 75% few parts at a lower cost than previous 1K and 2K chip designs. Static RAM technology increases reliability and reduces power consumption.

Configured in single, swing-out gate assembly, the 4135/4145 memory system is expandable in 256K byte increments to a maximum of 2 megabytes in conjunction with IBM. Memory increments can be installed or upgraded at a user site in just several hours, with mimimal processor change. No patch deck or floppy disk changes are required.

Additional features include automatic margining, which provides extremely low power dissipation when the device is not accessed; switching-regulated power supplies to provide wide power fluctuation tolerance and error checking and correcting.

A typical 256K byte increment of 4135/4145 memory is priced at $41,000. For more information contact Calcomp, Inc., 2411 W. La Palma Ave., Anaheim, CA 92801, (714) 821-2541, Carol Harris Felton.

CIRCLE INQUIRY NO. 140

Wyle uP/GPIB Compatibility

Wyle Laboratories/Computer Products now offers IEEE-488 GPIB compatibility for the Wyle uP microcomputer system.

This compatibility is via Wyle *software subroutines* that allow the standard Wyle BIO-2 buffered I/O module to function as a GPIB interface.

Users can no configure data acquisition and control systems using the Wyle uP microcomputer and existing GPIB compatible laboratory instruments.

The software is available on tape for $35 or EPROM for $95. For more information contact Wyle Laboratories/Computer Products, 3200 Magruder Blvd., Hampton, VA 23666, (804) 838-0122.

CIRCLE INQUIRY NO. 141

RAM-STOR 101/151 Memory Systems

A new RAM-STOR 101/151 memory system that can enhance NCR Century 101 computers by expanding memory capacity to 256K bytes and memory speed to 750ns cycle time increases speed by 37.5% and capacity by 100%. It effectively updates 101 processors to 151 capabilities at a savings of 30% to 40%.

RAM-STOR 101/151 memory mounted within the auxiliary cabinet includes gold plated contacts and worst case testing that combine to yield a reliable memory system. All memory systems are tested, burned in and retested with worst case parameters prior to installation.

 CIRCLE INQUIRY NO. 96

The RAM-STOR 101/151 is available at a cost of $45,000 for 128K bytes. For more information contact Computer Enhancement Corp., 3189-E Airway Ave., Costa Mesa, CA 92626, (714) 754-0521.

CIRCLE INQUIRY NO. 142

Safety-Approved Plotters

The Instruments Division of Gould Inc., announced that its complete line of electrostatic plotters and printer/plotters, the Gould 5000 Series, has been safety-approved by Underwriters Laboratories Specification 478 for Electronic Data Processing Equipment and authorized to carry the U.S. Listed Label.

The Standard for Electronic Data Processing Units and Systems, UL 478, is one of the more difficult U.L. classifications to meet. Equipment meeting the requirements of U.L. 478 is eligible for designation as N.F.P.A. Type II equipment in accordance with the National Fire Protection Association Standard for Protection fo Electronic Computer/Data Processing Equipment N.F.P.A. 75-1972.

Gould Electrostatic Plotters are used in a variety of scientific, engineering and business applications including brain and body scan, seismic recording, mapping, CAD, remote plotting, Pert charting, and CRT hardcopy. U.L. Safety-Approval has become a major factor in all of these application areas.

For more information contact Marketing Services, Gould Inc., Instruments Div., 3631 Perkins, Ave., Cleveland, OH 44114, (216) 361-3315.

CIRCLE INQUIRY NO. 143

Push Buttons

Square push buttons, widely accepted in the computer and electronics industry for their attractive and space-saving shape, are now made by General Electric for rugged, industrial applications.

General Electric's new line of compact industrial square oiltight push buttons and indicating lights have been designed to give continuing performance where oil, coolants and other contaminants are present.

Push buttons are available in black, red, yellow, green, white and blue, and indicating light lenses may be specified in white, red, amber and blue. For more information ont he new GE CR104M square industrial oiltight push buttons and indicating lights contact General Electric Co., General Purpose Control Dept., P.O. Box 2913, Bloomington, IL 61701.

CIRCLE INQUIRY NO. 144

Futra Model-10 Line Printer

The Futra Model-10 Line Printer provides high quality character printout for terminals and mini/microcomputers. Housed in an attractive desk top enclosure, the Model-10 incorporates a belt impact, full-character (ot dot matrix), 80 column printing mechanism.

The Model-10 operates at a minimum print rate of 150 lines per minute, using the 64 ASCII character set or a minimum of 84 lpm using a full 96 ASCII character set and can produce up to four copies including the original.

The Model-10 is priced at $2695.00 and comes standard with a pin feed paper handling mechanism, format control unit (top of form), either 64 or 96 ASCII character set and parallel interface. For more information contact Futra Co., 3421 Onyx St., P.O. Box 4380, Torrance, CA 90510, (213) 371-8138. Dealer inquiries invited.

CIRCLE INQUIRY NO. 145

EX3000 Computer Systems

The Extensys EX3000 microprocessor-based computer systems provide exceptionally high performance through distributed processing techniques common to large scale computers. Distributed processing allows for the distribution of system activities, computational and input/output operations to the hardware components that are best equipped to perform them.

The hardware of the EX3000 Computer Systems consists of subsystems and p/c board components which communicate via a common bus structure.

Each of these subsystems contain multiple microprocessors to implement the distributed processing of the EX3000 Systems.

For more information contact Extensys Corp., 380 Bernardo Ave., Mountain View, CA 94040, (415) 969-6100.

CIRCLE INQUIRY NO. 160

Check Feature Protects Against Counterfeiting

NCR Corporation has announced a printing technique which prevents the use of color copiers in counterfeiting checks and other valuable documents by causing the word "COPY" to appear on the face of the copied document.

"Stop-a-Copy" is, in effect, a built-in automatic alarm system on the face of each check. In the original check the word "COPY" blends with the colored background on the check and is not clearly visible to the unaided eye. However, when a duplicate is produced by a color copier, the background fades out, leaving the word "COPY" clearly visible to those who might otherwise cash the counterfeit check.

The "Stop-a-Copy" printing process can be used with payroll checks, money orders, voucher checks, gift certificates, dividend checks, and cashier checks.

For more information contact NCR Corp., Systemedia Div., Dayton, OH 45479, (513) 449-2150.

CIRCLE INQUIRY NO. 146

SuperDEC

The new SuperDEC Throughput Optimizer is a printed circuit board designed to replace the existing digital electronics in Digital Equipment's DECwriter II teleprinter. DEC users can pull out the guts of their DECwriters and screw in the brains of the SuperDEC Optimizer.

Standard features include automatic and manual top-of-form, full horizontal and vertical tabs (addressable and absolute), adjustable right and left margins and an RS-232C interface. SuperDEC carries a full one year warranty on all parts and workmanship.

Price is $395. All deliveries are F.O.B. the factory in Charlotte, NC. Delivery is 15-30 days ARO. For more information contact Intertec Data Systems, 1851 Interstate 85 South, Charlotte, NC 28208, (704) 377-0300.

CIRCLE INQUIRY NO. 159

PROM Programmer

The PP-2708/16 PROM programmer plugs directly into any 2708 or TMS 2716 memory socket. Simply drop a PROM into the zero insertion force socket and a short software routines sends the data over the eight lower address lines using a unique interfacing technique. No additional power supplies are required and all timing and control sequences are handled by the programmer. In addition, multiple programmers may be connected in parallel for gang programming.

Each programmer comes complete with a DC to DC switching regulator, ten turn cermet trimmers for precise program voltage and pulse width alignment, and a zero insertion force socket. The unit is packaged in a handsome black anodized aluminum case for table top operation. A 5-foot ribbon cable terminated with a 24-pin plug connects the unit to the read only PROM socket. (The programmer may also be interfaced to an 8-bit parallel port.)

The Model PP-2708/16 programs 2708s and TMS 2716s; the Model PP-2716 programs the unique Intel 2716 EPROM. Both sell for $295. For more information contact Oliver Advanced Engineering Div., 676 W. Wilson Ave., Glendale, CA 91203, (213) 240-0080.

CIRCLE INQUIRY NO. 161

Smart EPROM Gang Programmer

The SMI-800 is a microprocessor-based EPROM programmer featuring extensive user functions. The SMI-800 programs up to eight EPROMs simultaneously by switch selection of the desired device type: 2704s, 2708s, Intel's 2758s, TI's 2716 and Intel's 2716s. The powerful editor allows the user to display and change the contents of a program, plus the user can move memory around and insert new data.

The SMI-800 is designed to operate as a stand alone system, with a terminal, or as a computer peripheral. The SMI-800 has an RS 232C and 20mA current loop interface built in.

The SMI-800 price is $3850. The paper tape reader option is $575. Delivery is immediate. For information contact Shepardson Microsystems, Inc., 20823 Stevens Creek Blvd., Bldg. C4-H, Cupertino, CA 95014.

CIRCLE INQUIRY NO. 162

Axiom Micrographics Printer

The EX-820, from simple user commands, can mix high resolution graphics and full ASCII alphanumerics. The printer, driven by an Intel 8048, is available from Axiom at a single quantity price of $795.

Under software control, users have unlimited flexibility in mixing alphanumeric ASCII fields and graphic fields on any line. The user can define the size of each graphic field, and can choose from four pre-programmed horizontal dot resolutions up to 128 dots per inch.

The EX-820 is a complete, stand-alone printer/plotter including case, power supply, parallel ASCII and RS232C/20mA serial interface, character generator, low paper detector, bell, built-in self tester and paper roll holder.

Delivery is 30 days ARO. For more information contact Simon Harrison, Vice President of Marketing, Axiom Corp., 5932 San Fernando Rd., Glendale, CA 91202, (213) 245-9244.

CIRCLE INQUIRY NO. 163

Axiom Intelligent Line Printer

A new intelligent electrosensitive line printer, model EX-801 MicroPrinter, with a host of features not available on other printers, and a sleek molded case style by INOVA design group, is now in production at Axiom.

This ultra-compact desk-top printer operates at up to 160 characters per second and offers users the choice of three character sizes to provide 80, 40 or 20 columns on the 5-inch wide electrosensitive paper. It is ideally suited for any application needing fast, low-cost hardcopy.

The Axiom EX-801 MicroPrinter is a complete, stand-alone printer including case, power supply, parallel ASCII and RS232C/20mA interface, character generator, low paper detector, bell, built-in self tester and paper roll holder.

The EX-801 is priced at $655 in unit quantities. $450 at 100 unit OEM. Delivery is 21 days ARO. For more information contact Simon Harrison, Vice Pres. of Mktg., Axiom Corp., 5932 San Fernando Rd., Glendale, CA 91202, (213) 245-9244.

CIRCLE INQUIRY NO. 164

BOOK REVIEWS

HOME COMPUTERS: 2^{10} QUESTIONS AND ANSWERS Volume 1: Hardware

By Rich Didday, dilithium Press

Review by Roger H. Edelson
Hardware Editor

I think I should start this review off with an apology — I am really not partial to Rich Didday's writing. Of the three books written by him that I have reviewed, I really haven't like any; maybe I'm a scholastic snob — or something. Be that as it may, I'll try to give a reasonably objective review of this book.

This book is written in the form of a dialog between one person with a substantial background in computing and another interested beginner. This procedure has worked very well before. However, Didday's attempt to make it a question and answer session results in an artificial breakup of the dialog. In many cases, the questions should not even have been given a number, and in others, the answers are worthy of more than one number. It evens out in the end, but the dialog becomes stilted and artificial through the use of numbers.

The book is divided into days (presumably that's all the Q's that A can take at one time) rather than chapters. In Day One, the book presents a wide-ranging overview of the microcomputer scene. Topics ranging from buzz words, organization of computer systems, through high-assembly and higher-level languages are covered.

In Day Two: Numbers, Logic, and Building Blocks. Boolean algebra is covered; however, De Morgan's theorem is introduced two pages too late to save us from a very wordy attempt to describe the equivalence of $A + B$ and $\overline{A} \bullet \overline{B}$. If the author really felt De Morgan's theorem should wait, why not at least use a truth table? By Day Three we are getting into hardware with some block and circuit diagrams. A good discussion of tri-state busses is included along with a cursory introduction of interrupts. I think the section on state-transistion diagrams could have been left out, and the small effort to compare wire-wrap, and printed circuit boards should have gone in the next day.

Day Four gives you some idea of what it's like to assemble your own microcomputer system. The discussion of the 8080-based machine did not really provide much substance, but the portion dealing with the Sphere machine might tend to frighten off all but the most stout-hearted. As one of the unfortunate few who own a Sphere system, I felt Didday was a little too easy on them — for one, he didn't mention the keyboard problem, or the horrible card interconnect system. Luckily, most kits do not provide all those headaches.

I think it would have been a little better if the book had indicated that the build-it-yourself approach does provide a little more familiarity with the system, and it makes repairs much easier. This portion of the book also attempts to cover some of the equipment available to the home computerist — like any attempt to list items in a rapidly changing marketplace the list is only valid the day it was written, but it does provide an inkling of what's out there and the price ranges.

Day Five begins the discussion of specific microprocessors, namely the 8080 and the 6800. The functional organization is compared and some of the instructions are investigated. Appendices cover the 6800 and 8080 instruction set, and the ASCII character set.

All in all, the book does cover a lot of ground, and the conversational approach does allow it to do so without the hindrance of formal structure. However, a little more editing could have been used. □

BEGINNING BASIC

By Paul M. Chirlian
dilithium Press

Review by Roger H. Edelson
Hardware Editor

The introduction to this book states it is designed as an introductory text on the BASIC programming language at the high school level or higher. It is intended for students who have essentially no experience with either computer programming or computers. It is written as a standard textbook and therefore differs greatly from the conversational interactive approach used by other beginning BASIC books. Also this text does not absolutely require access to a computer, (but there are exercises to be done on a computer), and does not encourage the "hand-on" try-it-and-see-what-happens method that is the hallmark of the "My Computer Likes Me . . ." style. While the subject is covered in reasonable detail, and adequate exercises are provided, the student is left with a more formal stand-offish impression of BASIC programming.

Arithmetic operations are presented first, followed by input and output statements, and in Chapter 4 the control statements. Throughout the text, the reader is cautioned to be aware of the variations that occur between different versions of BASIC. The author, however, does not identify the parentage of the BASIC he is using. Also, it would be nice if he had identified the acronym BASIC (Beginners All-purpose Symbolic Instruction Code), and told us a modicum of its history.

After we have become familiar with the fundamental BASIC operations the book introduces the concept of loops. Nested loops are discussed and some rules formulated for their use. The author again cautions the student to be aware of the degree of nesting allowed by the particular BASIC being used. This warning does not appear very prominently, and it would also be nice if the problems limiting unrestricted nesting are mentioned. After a grounding in loops, the book covers arrays and subprograms.

Chapter 8 covers the manipulation of alphanumeric characters using string variables. Unfortunately, the chapter is much too brief, and the useful (but difficult) concepts of pulling a string apart (i.e., Left \$, Mid \$, etc.) are left too brief and incomplete. The final two chapters on Vector/Matrix operations and Data Files are well written and provide good coverage of the subject material.

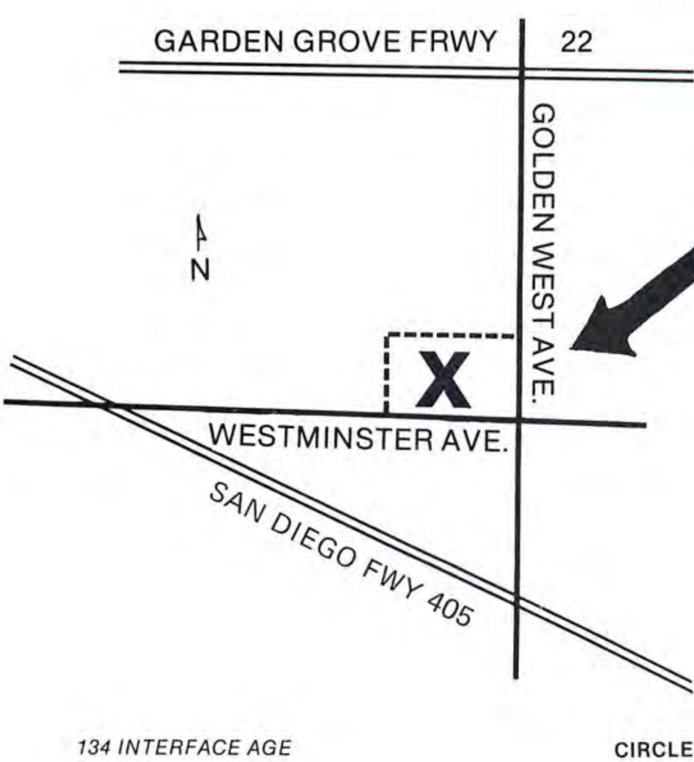
The book is written in a readable style and the exercises provide adequate examples to test your knowledge. I have to disagree with the blurb on the rear cover of the book that nothing is left out; as mentioned previously, I find some inexplicable gaps. Also, while the book will surely teach you how to program in BASIC, contrary to the rear cover blurb, it will not really make "a good BASIC programmer of any reader." □

OPENING SOON:

COMPUTER COMPONENTS OF ORANGE COUNTY

(FORMERLY COMPUTER PLAYGROUND)

- PRODUCTS TO BE CARRIED: IMSAI, VECTOR GRAPHICS, APPLE, TDL, POLYMORPHIC, PROCESSOR TECHNOLOGY, CROMENCO, PERSCI, CENTRONICS, TARBELL, VISTA, SHUGART, NORTH STAR, HURISTICS, COMPUTER WORLD, KOYO MONITORS, KIM, MICRO-DESIGN AND MANY MANY MORE . . . . . .
- OUR NEW LOCATION IS NEXT DOOR FROM WHERE WE PRESENTLY ARE.
- WE WILL BE THE **LARGEST** (OVER 5000 SQUARE FEET) AND **MOST COMPLETE** COMPUTER STORE IN SOUTHERN CALIFORNIA.
- BROWSE THROUGH THE **LARGEST MICROCOMPUTER BOOK AND MAGAZINE COLLECTION** ON THE WEST COAST.
- WE ALSO **TEACH CLASSES IN BASIC AND ASSEMBLY LANGUAGE** AS WELL AS **PROVIDE SERVICE AND REPAIR.**
- WATCH FOR OUR **GRAND OPENING BONANZA SALE.** COME AND VISIT OUR STORE AND SAVE **$$$$$$$$$** . . . .



Computer Components of Orange County
6791 Westminster Ave.
Westminster, Calif. 92708
(714) 898-8330

Other Computer Component Stores Located at:

- 5848 Sepulveda Blvd.
  Van Nuys, CA 91411
  (213) 786-7411
- 4705 Artesia Blvd.
  Lawndale, CA 90260
  (213) 370-4842

CIRCLE INQUIRY NO. 82

Cumulative Author Index to INTERFACE AGE

December 1975 through December 1977

December 1975-July 1976 issues were published under the name of SCCS INTERFACE. All issues following July 1976 are under the name of INTERFACE AGE.

AUTHORS . . . IN ALPHABETICAL ORDER BY LAST NAME
Format: Month/Year Volume/Issue Title Page

An Interrupt Driven Floppy Disk Controller for the S-100 Bus

By Robert Wright and Chester P. Quinn

The recent announcement that National Semiconductor will cooperatively second source the Western Digital FD1771 Floppy Disk Formatter/Controller substantiates the probability that the device will become a readily available, economical industry standard; even its present pricing (about $60.00, quantity one) makes the bottom line price of an IBM 3740 compatible floppy system virtually dependent on the cost of the drive and its power supplies, an amount that is certainly within the reach of a large group of potential users.

This article describes an S-100 bus compatible interrupt driven floppy disk controller based on the FD1771 and which conforms to the 3740 format. The fact that the controller is 3740 compatible is a software driven parameterization, and simple modification to the driver routines allows one to readily obtain on the order of 100K bytes more storage than the 250K bytes afforded by the IBM standard. The FD1771 is possessed of enough intelligence to be configured in a number of ways, and the initial key to an understanding of this design is a brief description of the IC itself.

Figure 1 shows the pinouts of the forty-pin DIP, which is available in both ceramic and plastic, the ceramic part being distinguished by an A suffix, the plastic by a B. A block diagram of the IC is illustrated in Figure 2. The primary sections of the device are the processor interface and the floppy drive interface, which are comprised of functional blocks described by Western Digital as follows:

Data Shift Register (DSR) — This 8-bit register assembles serial data from the read data input (FDDATA) during read operations and transfers serial data to the write data output (WD) during write operations.

Data Register (DR) — This 8-bit register is used as a buffer register during read or write operations. A data byte is transferred in parallel to or from the Data Shift Register for a write operation or a read operation respectively. This register also contains the address of the desired track when executing the *Seek* command. The Data Register can be loaded from the access lines (DAL0-DAL7) or read onto them under the control of the processor interface.

Track Register (TR) — This 8-bit register holds the track number of the current head position. It is automatically incremented or decremented for each step in (toward Track 76) or out (toward Track 00) respectively. Its contents are compared with the recorded track number in the ID field during read, write, or verify operations and with the contents of the Data Register during a seek operation. The Track Register may be loaded from or transferred to the DAL. It should not be loaded when the device is busy.

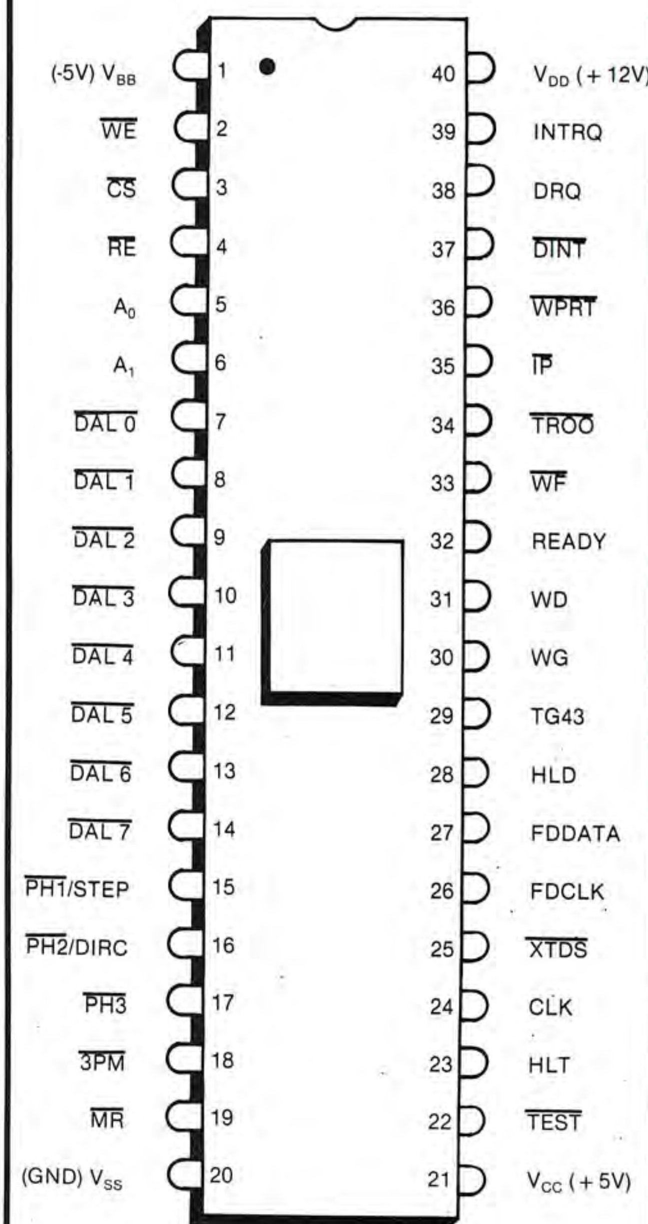


Figure 1. FD1771 Pin Connection Diagram.

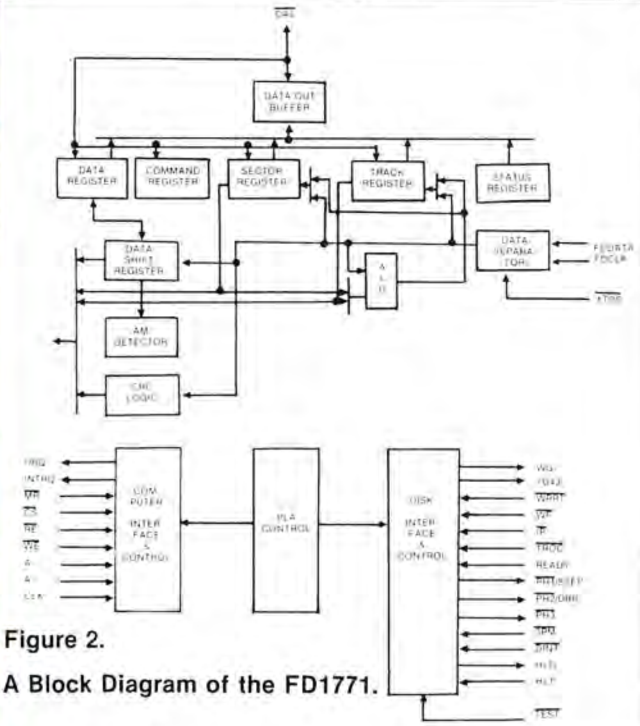


Figure 2.

A Block Diagram of the FD1771.

Sector Register (SR) — This 8-bit register holds the address of the desired sector position. Its contents are compared with the recorded sector number in the ID field during read or write operations. The Sector Register can also be loaded from or transferred to the DAL. It should not be loaded while the device is busy.

Command Register (CR) — This 8-bit register holds the command presently being executed. It should not be loaded while the device is busy unless it is desired to override the current command. This latter action results in an interrupt request (INTRQ). The Command Register can be loaded from DAL but not read onto the DAL.

Status Register (STR) — This 8-bit register holds the device status information. The meanings of the status bits are a function of the contents of the Command Register. The Status Register may be read onto the DAL but not loaded from them.

CRC Logic — This logic checks or generates a 16-bit Cycle Redundancy Check (CRC) character by the polynomial $G(x) = x^{16} + x^{12} + x^{5} + 1$. The CRC includes all information starting with the Address Mark (AM) and up to the CRC character. The CRC register is preset to ones prior to data being shifted through the circuit.

Arithmetic Logic Unit (ALU) — The ALU is a bit serial comparator, incrementer, and decrementer, and it is used for register modification and comparisons with the recorded ID field on the diskette.

AM Detector — This logic is used to detect ID, Data, and Index Address Marks during read and write operations.

Timing and Control — All interface controls to the host processor and the floppy disk drive are generated by this logic, which is based on a PLA. The internal device timing is derived from an external 2MHz clock.

The FD1771 can be considered a specialized microprocessor with an instruction repertoire of its own which consists of a set of eleven commands of four types that it will accept and execute. Command words should only be loaded into the Command Register when the Busy status bit is reset (off), with the exception of the Force Interrupt command used to terminate the present operation. Whenever the device is executing a command, the Busy status bit is set. When the command is completed, an interrupt request is generated and the Busy status bit is reset. The Status Register contents will indicate whether or not any error was encountered. A command summary is presented in Table 1, with the associated flags being summarized in Table 2. Table 3 contains the definitions of the status bits for various types of commands.

TYPE	COMMAND	BIT 7	6	5	4	3	2	1	0
I	[illegible]	0	0	0	0	h	V	r_1	r_0
I	[illegible]	0	0	0	1	h	V	r_1	r_0
I	[illegible]	0	0	1	u	h	V	r_1	r_0
I	[illegible]	0	1	0	u	h	V	r_1	r_0
I	[illegible]	0	1	1	u	h	V	r_1	r_0
II	[illegible]	1	0	0	m	b	E	0	0
II	[illegible]	1	0	1	m	b	E	a_1	a_0
III	[illegible]	1	1	0	0	0	1	0	0
III	[illegible]	1	1	1	0	0	1	0	s
III	[illegible]	1	1	1	1	0	1	0	0
IV	[illegible]	1	1	0	1	I_3	I_2	I_1	I_0

Table 1. FD1771 Command Summary

TYPE I

h = Head Load Flag (Bit 3)

h = 1, Load head at beginning
h = 0, Do not load head at beginning

V = Verify Flag (Bit 2)

V = 1, Verify on last track
V = 0, No verify

$r_0 r_1$ = Stepping Motor Rate (Bits 1-0)

$r_1 r_0$ = 00, 6ms between steps
$r_1 r_0$ = 01, 6ms between steps
$r_1 r_0$ = 10, 10ms between steps
$r_1 r_0$ = 11, 20ms between steps

u = Update Flag (Bit 4)

u = 1, Update Track Register
u = 0, No update

TYPE III

$\bar{s}$ = Synchronize Flag (Bit 0)

$\bar{s}$ = 0, Synchronize to AM
s = 1, Do not synchronize to AM

TYPE II

m = Multiple Record Flag (Bit 4)

m = 0, Single record
m = 1, Multiple records

b = Block Length Flag (Bit 3)

b = 1, IBM Format (128 to 1024 bytes)
b = 0, Non-IBM Format (16 to 4096 bytes)

$a_1 a_0$ = Data Address Mark (Bits 1-0)

$a_1 a_0$ = 00, FB data mark
$a_1 a_0$ = 01, FA data mark
$a_1 a_0$ = 10, F9 data mark
$a_1 a_0$ = 11, F8 data mark

E = Enable HLD and 10ms Delay

E = 1, Enable HLD, HLT and 10ms delay
E = 0, Head is assumed engaged and there is no 10ms Delay

TYPE IV

I_i = Interrupt Condition Flags (Bits 3-0)

I_0 = 1, Not Ready to Ready Transition
I_1 = 1, Ready to Not Ready Transition
I_2 = 1, Index Pulse
I_3 = 1, Immediate Interrupt

Table 2. FD1771 Command Flag Flag Summary

The *Restore*, *Seek*, and the three *Step* commands control the position of the read/write head over the desired track. The *Restore* command moves it over Track 00, the *Seek* command positions it over the track specified by the contents of the Data Register, and the *Step* commands position the head over a track adjacent to the present track. *Step Out* moves the head outward one track from the center (toward Track 00), *Step In* moves it inward one track toward the center (toward Track 76), and *Step* moves it one track in the same direction as the last command.

The *Read* and *Write* commands are those normally used in the transfer of data. The *Read* command initiates a search for a track and sector code in the ID field equal to the contents of the Track and Sector Registers. When it is found, the data following is converted from serial to parallel format and transferred to the Data Register on a

BIT	ALL TYPE I COMMANDS	READ ADDRESS	READ	READ TRACK	WRITE	WRITE TRACK
S7	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY	NOT READY
S6	WRITE PROTECT	0	RECORD TYPE	0	WRITE PROTECT	WRITE PROTECT
S5	HEAD ENGAGED	0	RECORD TYPE	0	WRITE FAULT	WRITE FAULT
S4	SEEK ERROR	ID NOT FOUND	RECORD NOT FOUND	0	RECORD NOT FOUND	0
S3	CRC ERROR	CRC ERROR	CRC ERROR	0	CRC ERROR	0
S2	TRACK 0	LOST DATA	LOST DATA	LOST DATA	LOST DATA	LOST DATA
S1	INDEX	DRQ	DRQ	DRQ	DRQ	DRQ
S0	BUSY	BUSY	BUSY	BUSY	BUSY	BUSY

Table 3. Status Register Summary

byte-by-byte basis with each transfer causing the data request (DRQ) to be set. Bit 4 in the *Read* (or *Write*) command may be set to allow multiple sector transfers, while Bit 3 may be set to select a sector byte count different than that of the standard IBM 128 byte format. When the ID field track and sector comparison is satisfied on a *Write* command, all words loaded into the Data Register will be transferred to the shift register on a byte-by-byte basis for parallel to serial conversion with each such transfer generating a DRQ signal. One of four separate Data Address Marks may be specified through Bits 0 and 1 of the *Write* command.

The *Read Address* command provides the next encountered ID field (six bytes) on the diskette to the processor. It may be used to determine the present track number and is particularly useful in a multiple drive environment.

The *Write Track* command is primarily used in formatting or initializing a diskette. Once the index mark is detected the device will request data and begin transferring it serially to the diskette. This data includes all ID fields, gaps, and Data fields. The values written for Address and Data Address Marks and for the CRC character are dependent upon certain data patterns presented to the device. The *Read Track* command allows the reading of the entire recorded pattern on a particular track, including the gaps, ID fields, and data sectors. The user should refer to the Western Digital data sheet for formatting details.

The final command is the *Force Interrupt* which may be loaded at any time to terminate the present operation. This command also allows for the selection of four conditions which will result in an interrupt request to the processor.

Execution of each of the command instructions will affect the value of the status bits contained in the Status Register. Bit 0 (Busy) and Bit 7 (Ready) will always indicate the status of the device concerning the present operation or the Ready condition line from the drive, respectively.

In general, Bit 1 will reflect the condition of the external DRQ line to the processor and Bit 2 will reflect a Lost Data condition due to an overrun or underrun in the data stream. The Type 1 or head positioning instructions uses Bits 1 and 2 to reflect the condition of the index pulse $\overline{\text{IP}}$ and Track 00 $\overline{\text{TR00}}$ inputs respectively.

Bit 3 is normally used to indicate that a CRC error was detected in the ID or Data fields, except during a *Read Track* or a *Write Track* command, in which case the CRC characters are not checked. Bit 4 indicates a failure to locate a desired track and sector. The value of Bit 6 reflects the write protect $\overline{\text{WPRT}}$ input on *Seek* and *Write* commands and is used in combination with Bit 5 during the *Read* command to identify the type of Data Address Mark encountered. Bit 5 is also used to indicate the head loading status on a *Seek* command and to indicate a write fault $\overline{\text{WF}}$ on the *Write* commands.

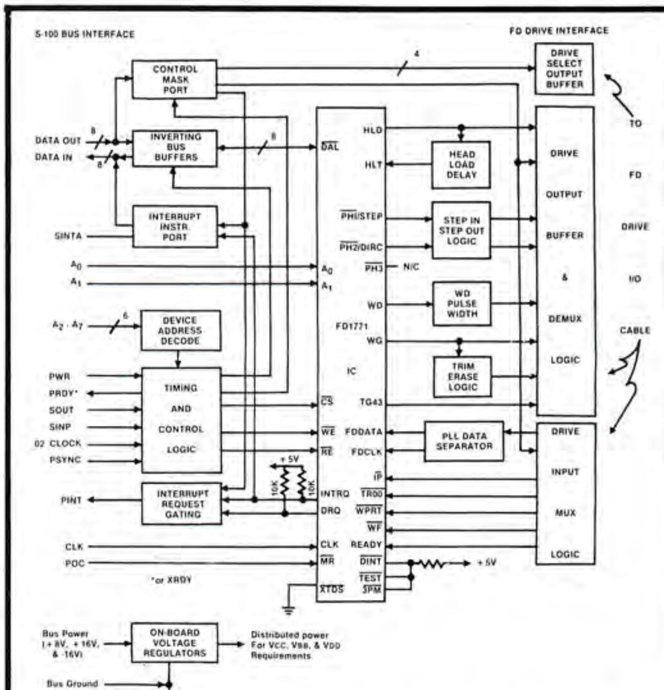


Figure 3. A Block Diagram of the S-100 Bus Floppy Disk Drive Controller.

A block diagram of the complete S-100 bus floppy controller is illustrated in Figure 3, while a schematic diagram of the actual logic is shown in Figure 4. An attempt was made at optimizing the flexibility of the interface hardware between the FD1771 and the disk drive itself so that drives from different manufacturers could be readily supported by the same controller with minimum modification. Consequently not all of the supporting integration may be necessary for a particular drive or application. The circuitry is shown with the jumper connections required to support a Pertec FD400 and the labeling of the interface lines to and from the drive reflects this manufacturer's nomenclature.

The FD1771 generates all of the control signals required to position the read/write head over the desired track. It has the capability of sending successive three-phase pulses over lines PH1, PH2, and PH3 or of sending a level over the PH2 line (which becomes DIRC) and pulses over the PH1 line (which becomes STEP) for drives using a step-direction motor. The particular mode of operation is determined by hardwiring an external control pin, 3PM, low (= 0) for the former, and high (= 1) for the latter. IC U16B is provided should it be necessary to modify the width of the step pulse from the FD1771, IC U24, which is nominally 4μs wide. IC U17B and C together with U18C and E form a one-of-two decoder to provide the separate "Step In" and "Step Out" lines required by the Pertec drive. As may be seen from Table 2, the stepping rate is specified by the command word through Bits 0 and 1.

The head is loaded against the diskette by the HLD (head load) signal from the FD1771. No read or write operation may occur until a logic high is sampled at the HLT (head load timing) input. This input is sampled after a 10ms internal delay and may be wired high if this time is sufficient or an external one-shot, IC U12B, may be used to extend this time. If the head is already engaged from a previous operation, the resetting of Bit 2 in the *Read* or *Write* command will disable the HLT function and the 10ms delay.

The $\overline{\text{IP}}$ (index pulse) and $\overline{\text{TR00}}$ (Track 00) outputs from the drive indicate when the index mark has been encountered (once per revolution) or when the read/write head is over Track 00 respectively. These signals are presented to the FD1771 in an essentially unmodified form, as are the $\overline{\text{WPRT}}$ (write protect), $\overline{\text{WF}}$ (write fault), and Ready status lines. The write protect signal, when low, will prevent the FD1771 from executing a *Write* command. The write fault signal, when low, signifies a write operation fault in the drive, such as failure to detect write current when the write gate (WG) or write enable signal is on, and will result in a termination of the current *Write* command. The Ready input signal indicates the readiness of the floppy drive, and a logic low on this input will prevent any *Read* or *Write* command from being executed. The disk initialization input ($\overline{\text{DINT}}$), when low, will prevent the execution of the *Write Track* command and essentially disables the rewriting of a format over a previously formatted diskette. For most applications, particularly those in which the user may wish to preformat a diskette, this input may be tied high as shown in the schematic or connected to one of the spare outputs (Bit 5 or 6) of IC U25 (8212).

The write gate (WG) or "write enable" output signal from the FD1771 is activated to allow current to flow through the drive's read/write head. IC's U13A and B, U14A, and U17A are used to produce the "trim erase" signal required by the Pertec FD400. This signal goes true (high) at U14A, pin 6 ($\overline{\text{Q}}$), some 200μs after the beginning of WG and returns false about 475μs after the end of the WG signal and is used to erase a "guard band" between data tracks to improve media interchangeability and to lower track-to-track crosstalk. Most newer drives provide this feature internally.

Two other outputs associated with write operations connect to the drive from the FD1771. One of these is the WD (write data) output, which is a serial bit stream consisting of interleaved clock and data pulses, each nominally 500ns wide. A one-shot, IC U16A, is provided to modify the width of these pulses if necessary. The other is the TG43 (track greater than 43) signal to the drive indicating that the track to be written on is between Track 44 and Track 76. This output will cause the drive to lower the write current on these inner tracks to compensate for the higher bit density.

The serial data read from the floppy drive may be input as composite data (unseparated clock and data) to the FDDATA (floppy drive data) input or as separated data in which the data is input to the FDDATA pin and the recovered clock is input to the FDCLK (floppy drive clock) pin. This latter mode of operation, which requires an external data separator circuit, is recommended by Western Digital and is obtained by grounding the XTDS (external data separator) input pin on the FD1771.

A phase locked loop (PLL) clock and data separator recommended by Motorola is shown in the schematic. The MC4024 voltage controlled multivibrator, IC U23, supplies an internal clock frequency of 8MHz which is thirty-two times each bit cell time. IC U21 (74LS161) divides this clock frequency by sixteen and provides a carry pulse to one side of IC U22, an MC4044 phase detector.

The incoming raw data is first shaped by the Schmitt trigger NAND gate, IC U9A or B (7413), and then fed to two flipflops, IC's U15A and U14B, which generate a pulse whose width is equal to one period of the 8MHz PLL clock, or 125ns, for each data or clock pulse received from the drive. These pulses are stretched to about 250ns by a one-shot, IC U12A, and fed to IC gates U11A and U11B, where they are NANDed with a "data window" or a "clock window" respectively, to provide separated clock and data outputs.

These separator windows are formed by counter IC U20 (74LS161) which clocks the flipflop U15B once for every eight counts of the PLL clock to provide a 50% window every 2μs. Each time a bit is received in the data stream from the drive, the $\overline{\text{Q}}$ output (pin 8) of IC U14B will pulse low to load a count of nine into this counter. If the PLL clock frequency is correct, this will not alter the count. If, however, the clock is not concurrent with the input pulse, the count will be advanced or set back to alter the window and change the timing of the carry output pulse. This carry pulse is then applied to the other input of IC U22, the MC4044 phase detector mentioned above. The output of the phase detector is a signal proportional to the phase or frequency difference between the incoming data pulses and the system clock. This signal is fed through a low pass filter for stability to the input of IC U23, the MC4024, to control the system frequency.

Counter IC U19 (74LS161) and flipflop U15B above act to provide the proper phase relationship for the separated data and clock pulses. This is accomplished by looking for missing clock and data pulses and if no clock pulse is detected within four transitions of flipflop U15B, the Q_C output of U19 will cause the flipflop to reverse the sampling windows.

It is interesting to note that a test program used by the authors to gauge the separator's data recovery reliability continuously wrote and then read the entire surface of a diskette for more than ten hours without a detected error, which seems to well justify the small additional component costs associated with this circuit.

The IC's U1A-D, U2A-D, U3A, U4A-C, and U5A-C are open-collector NAND gates (74LS03's or 7403's) used to buffer the signal lines to the drives. These gates are arranged

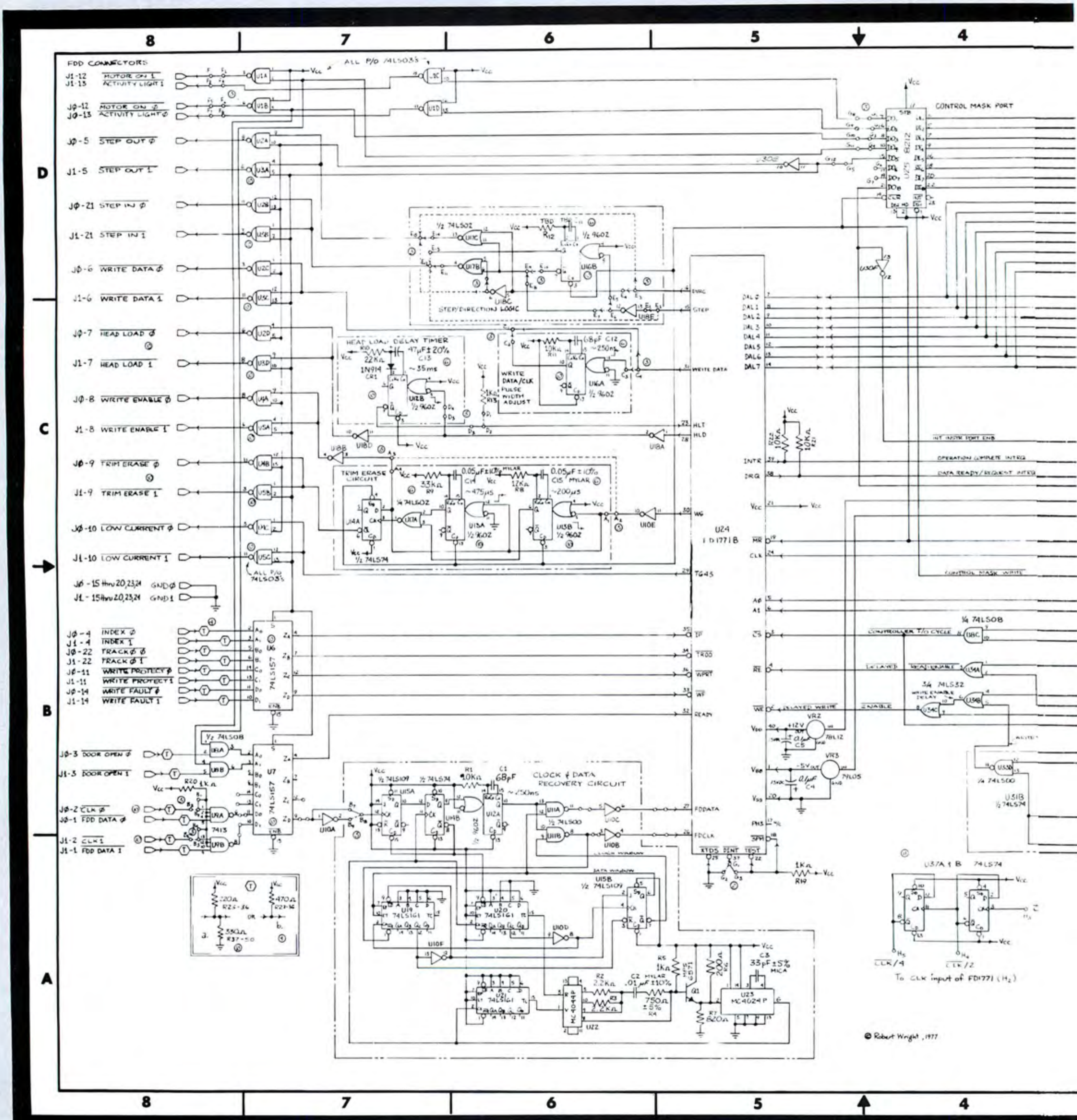
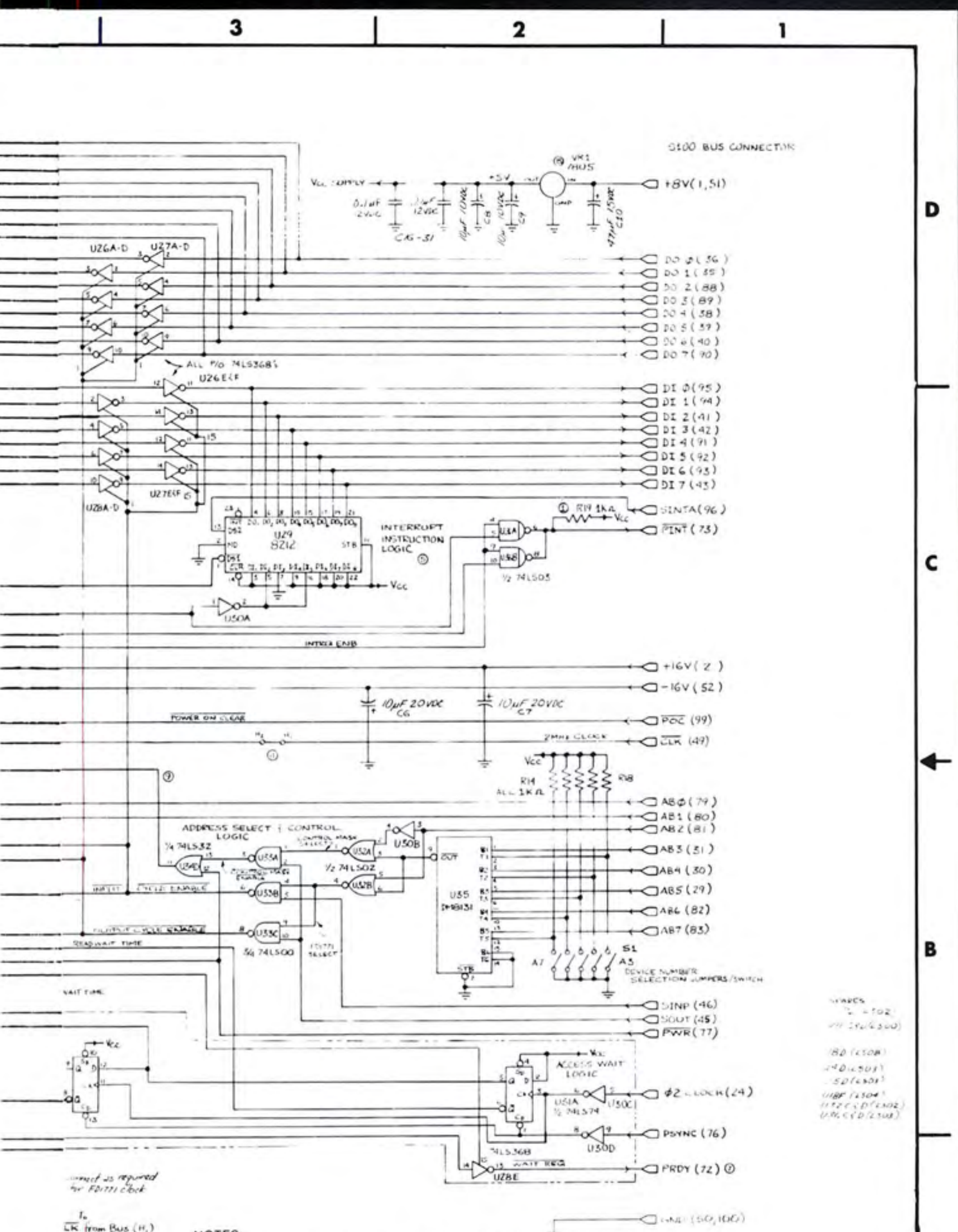


Figure 4. Schematic of FD1771.



NOTES:
All polarized capacitors are tantalum, non-polar ceramic unless otherwise specified.

1. Not required for Altair CPU Card.
2. DINT must be high to format disks.
3. Jumper as requires for drive used. Shown for FD400.
4. Terminate as required for drive used. Use a. for FD400.
5. Jams 'EI' for DRQ, 'PCHL' for INTR if enabled.
6. Value depends on drive used. Actual values shown typical for FD400.
7. If front panel wait request to XRDY (3), else use XRDY (3).
8. Mount on heat sink; thermalloy #6106-13.
9. Active with any of upper four device numbers from block of eight selected by S1. FD1771B selected by lower four device numbers.
10. Not required for all applications.
11. For applications where CLK (S-100 pin 49) is 4MHz or for use with minifloppy drives.

PARTS LIST for the S-100 Floppy Controller:

****Capacitors	
C1	68pF ceramic
C2	0.01uF ±10% mylar
C3	33pF ±10% mica, CM05ED330J or equiv.
C4,5	0.1uF 15V tantalum
C6,7	10uF, 20V tantalum
C8,9	10uF, 10V tantalum
C10	47uF, 15V tantalum
C11	value as req'd.
C12	value as req'd., 68pF shown
C13	value as req'd., 47uF ±20% tantalum shown
C14,15	value as req'd., 0.05uF ±10% mylar shown
C16-31	0.1uF 16V ceramic
****Resistors	(Resistance in Ohms, All are ±10% and ¼Watt unless otherwise shown)
R1,21,22	10K
R2,3	2.2K
R4	750 ±5%
R5,13-20	1K
R6	200 ±5%
R7	820
R8	value as req'd., 12K shown
R9	value as req'd., 33K shown
R10	value as req'd., 22K shown
R11	value as req'd., 10K shown
R12	value as req'd.
R23-36	value as req'd., 220 shown
R37-51	value as req'd., 330 shown
****Integrated Circuits	
U1-5,36	74LS03 Quad two-input open collector NAND
U6,7	74LS157 Quad two-input multiplexer
U8	74LS08 Quad two-input AND
U9	7413 Dual 4-input Schmitt trigger NAND
U10,18,30	74LS04 Hex inverter
U11,33	74LS00 Quad two-input NAND
U12,13,16	9602 Monostable
U14,31,37	74LS74 Dual D-type flipflop *SEE SCHEMATIC FOR SPARE FUNCTIONS*
U15	74LS109 Dual JK flipflop
U17,32	74LS02 Quad two-input NOR
U19-21	74LS161 Presettable binary counter
U22	MC4044P Phase detector
U23	MC4024P Dual voltage-controlled multivibrator
U24	FD1771B Floppy disk Formatter/Controller
U25,29	8212 8-bit multimode latch
U26-28	74LS368 Inverting hex bus buffers
U34	74LS32 Quad two-input OR
U35	DM8131 Hex unified bus comparator
****Miscellaneous	
CR1	1N914 or 1N4148, Silicon diode
Q1	MPS6571 NPN low noise silicon xstr.
VR1	7805 +5V regulator
VR2	78L12 +12V regulator
VR3	79L05 -5V regulator
Misc.	TO-220 heat sink, Thermalloy 6106B-13 or equiv.

inpairs to form 'one-of-two' demultiplexer circuits to allow the use of floppy drives which require a separate connecting cable for each drive. The controller will support up to two drives of this type when wired as shown in the schematic. In this case, the "drive select" function is determined by the value of Bit 4 written by an 'OUT' instruction into the control mask port, IC U25 (8212). Bits 0 through 3 from this port are buffered to the two drives as "motor on" and "activity light" control lines through IC U1. For drives which provide a multiplexed cable "daisy chain" capability so that all of the drives may be paralleled on a common cable, one NAND gate from each of the demultiplexer channels may be eliminated. If the drives require a device select line or a device select code, Bits 0 through 3 from IC U25 may be used together with IC U1 to provide up to four such separately controllable lines to the drives. The user should consult the particular drive's manual for specific details on its requirements.

Whenever the host system requires access to any one of the floppy controller's registers it must execute an 'IN' or an 'OUT' instruction. A block of eight device addresses associated with the controller is decoded by IC's U35, a 6-bit bus comparator (DM8131), NOR gates U32A and B (74LS02) and inverter U30B, using the bus "ADDRESS" lines A2 through A7. The values of the upper five bits (A3 through A7) of the controller's base address may be selected by jumpers or a switch at position S1.

Address bit 2 (A2) is wired on the controller board through IC U32A together with U30B in such a way as to access the control mask port if A2 is true (high) by making the U32A output high. This output, control mask select, is NANDed with the "SOUT" output cycle status line from the bus by IC U33A (74LS00) whose active low output, control mask enable, is then ORed with the $\overline{PWR}$ (processor write) bus control line by IC U34D (74LS32). This gate then generates the active low "control mask write" signal which actually causes the information on the bus "DATA OUT" lines to be written into the control mask port, an 8212 (U25), by pulsing its $\overline{DS1}$ input. The power-on-clear $\overline{(POC)}$ line from the bus is wired to its $\overline{CLR}$ input to reset all of its outputs low when power is first applied to the system.

Address bits A0 and A1 are not used in the selection of the control mask port and it occupies a block of four contiguous device numbers, any of which may be used to write into it as shown in Table 4. However, these least significant address bits are fed from the bus directly into the FD1771, and when combined with the $\overline{CS}$ and $\overline{RE}$ or $\overline{WE}$ signals caused by the execution of an 'IN' or an 'OUT' instruction, will be interpreted as selecting one of five accessible registers as shown in Table 4.

All commands, status, and data to and from the controller chip itself are transferred in inverted form over the tri-state bidirectional data access lines. The "negative true" logic used on these eight lines $\overline{(DAL0\text{-}DAL7)}$ effectively suggests the use of inverting buffers (74LS368's) in both directions to interface them easily to the "positive true" bus. The $\overline{DAL}$ are enabled as output drivers when both the $\overline{CS}$ (chip select) and $\overline{RE}$ (read enable) inputs are active (low) and are buffered to the S-100 bus "DATA IN" lines by IC's U26E and F, U27E and F, and U28A-D. They act as input receivers when both the $\overline{CS}$ and $\overline{WE}$ (write enable) inputs are low and are then driven from the bus "DATA OUT" lines by IC's U26A-D and U27A-D.

As may be seen from the schematic, if the A2 address bit is false (low) when the controller's base address as selected by bits A3 through A7 at S1 is decoded by IC U35, the output of NOR gate U32B will become high. This output, FD1771 select, is applied to NAND gate U33B together with the "SINP" input cycle status line from the bus to generate the active low "input cycle enable" signal and to NAND gate U33C together with the "SOUT" output cycle status line to generate the "output cycle enable" signal, also active low. The presence of one of these two signals indicates that an in-

POSITION S1: BASE ADDRESS SELECT

	To select a BIT value of	0 (zero) or	1 (one) for
(MSB)	A7, Pins 7 and 14 are	shorted	open
	A6, Pins 6 and 13 are	shorted	open
	A5, Pins 5 and 12 are	shorted	open
	A4, Pins 4 and 11 are	shorted	open
(LSB)	A3, Pins 3 and 10 are	shorted	open

These address bits select a contiguous block of eight device numbers which are:

Address Bits A2	A1	A0	Type of Access INPUT	OUTPUT
0	0	0	Status Register, FD1771*	Command Register, FD1771*
0	0	1	Track Register, FD1771	Track Register, FD1771
0	1	0	Sector Register, FD1771	Sector Register, FD1771
0	1	1	Data Register, FD1771	Data Register, FD1771
1	0	0	NONE	Control Mask Port, 8212
1	0	1	NONE	Control Mask Port, 8212
1	1	0	NONE	Control Mask Port, 8212
1	1	1	NONE	Control Mask Port, 8212

*The Status Register cannot be Output to and the Command Register cannot be Input from

Table 4. Floppy Controller Device Number Assignments.

put or output cycle is in progress and they enable, respectively, the read output or write input buffers described in the previous paragraph. Since one or the other of these signals will be low during any I/O operation involving the FD1771, they are ANDed together by IC U8C (74LS08) to generate the active low "controller I/O cycle" line used as chip select ($\overline{CS}$) by this device. The constraints of the data and address set up and hold times before, during, and after the $\overline{RE}$ and $\overline{WE}$ signals to the FD1771 require the use of some logic to incorporate one or more wait states when reading from or writing to it with the S-100 bus, and the "controller I/O cycle" signal is also used to enable the wait request buffer, U28E (74LS368). This output then controls the $\overline{PRDY}$ or the $\overline{XRDY}$ line on the bus when the controller chip is accessed. As noted on the schematic, only one of these two lines should be connected to the output of U28E.

D-type flipflops U31A and B (74LS74) are connected to form a 2-bit long shift register which is clocked by an inversion of the "Ø2 CLOCK" from the bus by IC U30C. This clock inversion insures the presence of stable information at the time the $\overline{PRDY}$ and $\overline{XRDY}$ lines are sampled. The shift register is cleared to all zeroes (both Q outputs low) by the occurrence of the PSYNC (processor sync) pulse on the bus near the beginning of each machine cycle. PSYNC, which is active high, is inverted by IC U30D to provide the active low level required by the flipflops' $\overline{C_D}$ (clear direct) inputs. The $\overline{Q}$ output of IC U31B, which is also at the end of the shift register, is connected to the input of the wait request buffer, U28E mentioned above. As a result, any I/O access to the four device addresses used by the controller chip from the S-100 bus will result in the insertion of two wait states into the processor's input or output machine cycles.

During the input machine cycle the $\overline{Q}$ output of flipflop U3A, read wait time, goes low after providing a delay of about one Ø2 clock cycle from the application of CS and a delay of about two clock cycles from the application of address bits A0 and A1 to the FD1771 IC. This "read wait time" signal is connected together with the active low "input cycle enable" signal discussed previously or to OR gate U34A (74LS32) to produce a "delayed read enable" signal for the controller IC's $\overline{RE}$ input. The next flipflop, U31B, inserts a second wait state beginning when the "delayed read enable" signal is made true (low) at the end of the first wait state. This signal will therefore be false (high) through the first three states (the T_1, T_2, and T_{W1} states for an 8080 system) and will then become true for the second wait state (T_{W2}) and continue thus through the next and final state (the T_3 or exit state for an 8080 system) of the input machine cycle.

During an output machine cycle, the Q output of flipflop U31A and the $\overline{Q}$ output of flipflop U31B are NANDed by IC U33D to generate an active low "write wait time" signal which is then ORed with the "output cycle enable" signal by IC U34B. This output, write enable delay, is then ORed with the low true $\overline{PWR}$ (processor write) signal from the bus by IC U34C to implement the "delayed write enable" signal applied to the FD1771's $\overline{WE}$ input. Therefore this line will be high (false) for the first three states (T_1, T_2, and T_{W1}) and then low for a second wait state (T_{W2}) in a manner similar to that of the "delayed read enable" signal described above. However, the "delayed write enable" line does not then continue low for the exit (T_3) state but instead is made to return high at the end of the second wait state, thereby providing an active signal to the $\overline{WE}$ input only during the third clock period (T_{W2}) of the processor's output cycle.

There are only two other input lines to the FD1771 from the processor interface. A 2MHz free running square wave clock is required by the chip as a reference for all timed signals such as motor control and data transfers. This signal is provided by applying the $\overline{CLK}$ line from the S-100 bus to the FD1771's $\overline{CLK}$ input pin. The $\overline{MR}$ (master reset) input, which is $\overline{POC}$ (power-on-clear) from the bus, clears the command register and initiates a *Restore* (see Track 00) command when the $\overline{MR}$ line returns to its inactive (high) state.

Finally, two signals are available to aid in program responses to the FD1771. Both of these are active high open drain type outputs and require a 10 Kilohm pull-up resistor to the Vcc supply. The INTRQ (interrupt request) signal is activated by the controller IC whenever an operation is completed successfully or terminated by a fault. The DRQ (data request) signal is activated as an indication of the chip's readiness to transfer a byte of data during read or write operations. Both of these signals respectively are used to activate the bus $\overline{PINT}$ (processor interrupt) line through open-collector NAND gates U36A and B (74LS03) if they are enabled by a high true "interrupt enable" signal from Bit 7 of the control mask port. This signal also enables the interrupt instruction port, IC U29 (8212), to jam an instruction onto the S-100 "DATAIN" lines in response to an interrupt acknowledge cycle status signal (SINTA) from the bus.

The interrupt instruction port is wired with inverter U30A in such a way as to jam an 'EI' instruction (373_8) onto the bus in response to a DRQ signal or to jam a 'PCHL' instruction (351_8) onto it in response to an INTRQ signal. This approach to interrupt driving the FD1771 takes advantage of its native intelligence and minimizes the complexity and length of the software routines necessary to support the controller.

To demonstrate the simplicity with which the FD1771 can be program controlled in an interrupt driven environment, let us consider the following subroutine which implements a "Read Sector" command for the controller:

```
;
; READ SECTOR COMMAND ROUTINE ENTRY POINT
;
                LXI     D, BEGADOR
                LDA     SECTOR
                OUT     SECTREG
                LXI     H, TAG2
                MVI     A, RDCMND
                OUT     CMNDREG
                EI
TAG1:           HLT
;
; AN 'EI' INSTRUCTION FOR A 'DRQ' SIGNAL OR A 'PCHL'
; INSTRUCTION FOR AN 'INTRQ' SIGNAL IS JAMMED ON THE
; DATA BUS IN RESPONSE TO AN INTERRUPT ACKNOWLEDGE
; CYCLE FROM THE PROCESSOR BETWEEN THESE INSTRUCTIONS.
;
                IN      DATAREG
                STAX    D
                INX     D
                JMP     TAG1
TAG2:           IN      STSREG
                ANI     ERRMSK
;
; READ SECTOR COMMAND EXIT POINT
;
```

The first six instructions set the number of the sector to be read from the diskette (the head is assumed to be over the correct track), place the destination address of the first byte from the sector into the register pair 'DE' and the address of the error check routine into register pair 'HL', and then orders the controller to perform a *Read* command. The processor then enters the 'HALT' state with interrupts enabled. When the first byte of data from the sector is assembled in the FD1771 and ready for transfer, the DRQ signal will generate an interrupt request to the processor. When the request is acknowledged, the controller will jam an 'EI' instruction into the processor which will then resume program execution with the 'IN DATAREG' instruction following the 'HLT' instruction in the listing. These next instructions get the

data byte from the controller, store it by the buffer pointer contained in the 'DE' register pair, and increment the pointer; whereupon the processor will jump back to the 'HLT' instruction at location TAG1 ready to acknowledge the next interrupt request and store the next data byte!

Since the controller IC will count off the proper number of DRQ's to transfer a sector, the only maintenance necessary for the buffer pointer is the increment instruction, 'INX D' in this example.

After the transfer of the last byte in the sector to the host system's memory, the processor will again be in the 'HALT' state with its interrupt request input enabled. At this point the FD1771 will generate an INTRQ signal causing an interrupt to the processor. When it is acknowledged, the controller will jam a 'PCHL' instruction into the processor which will then jump and resume execution with the sample error check routine at location TAG2.

A very similar sector write routine should exhibit exactly the same syntax and in fact would require substitutions for only three instructions; these being the 'MVI A, RDCMND' (change second operand to WTCMND), the 'IN DATAREG' (to 'OUT DATAREG'), and 'STAX D' (to 'LDAX D') instructions in the listing. In some small respect it could be said that since the controller determines what the host processor will do by inserting a variable instruction into an otherwise fixed program rather than being able only to cause it to begin execution at the location of some segment of totaly fixed code, the controller more nearly "programs the processor."

In the discussion on the "Read Sector" command it was stated that the head was assumed to be over the correct track. The following segment of code implements a routine which will first restore the head to Track 00 and then execute a seek to the specified track number and further demonstrates the capabilities and intelligence of the Western Digital FD1771.

```
;
; THIS SEGMENT IS THE RESTORE ROUTINE
;
RESTORE:    LXI     H, TAG3
            MVI     A, RSTRCMD
            OUT     CMNDREG
            EI
            HLT
;
; THE INTERRUPT REQUEST FROM THE CONTROLLER UPON
; COMPLETION OF THE RESTORE OPERATION WILL BE AC-
; KNOWLEDGED BY A 'PCHL' INSTRUCTION WHICH CAUSES
; EXECUTION TO CONTINUE WITH . . .
TAG3:       IN      STSREG
            ANI     ERRMSK1
;
; THIS SEGMENT IS THE SEEK ROUTINE
;
SEEK:       LDA     TRACK
            OUT     DATAREG
            LXI2    H, TAG4
            MVI     A, SEEKCMD
            OUT     CMNDREG
            EI
            HLT
;
; HERE THE 'PCHL' INSTRUCTION WILL CAUSE EXECU-
; TION TO CONTINUE WITH . . .
TAG4:       IN      STSREG
            ANI     ERRMSK1
```

A complete source listing of the 8080 system monitor program used by the authors to support the popular disk based CP/M software package available from Digital Research is on the following pages. With this monitor the user need only mount any CP/M based system diskette on the drive. The program then interrogates the diskette to determine the memory size for which it was initialized. If this value is less than or equal to 32K bytes of storage, it will then bootstrap any such diskette *without in any way modifying it* by providing for linkages to the BIOS program already resident on the CP/M diskette. This approach allows diskettes which were generated on any system with the same or smaller amount of memory to be bootstrapped and executed through the monitor and then returned unmodified (at least insofar as the BIOS is concerned) to the generating system, thereby eliminating the necessity for an otherwise unavoidable BIOS reinitialization of the diskette for each such exchange.□

TOP OF PC BOARD		BOTTOM OF PC BOARD	
1	+5V	51	+5V
2	+15V	52	-15V
3	XRDY	53	SSW DSB
4	VI 0	54	EXT CLR
5	VI 1	55	
6	VI 2	56	
7	VI 3	57	
8	VI 4	58	
9	VI 5	59	
10	VI 6	60	
11	VI 7	61	
12		62	
13		63	
14		64	
15		65	
16		66	
17		67	
18	STAT DISABLE	68	MWRITE
19	C/C DISABLE	69	PS
20	UNPROTECT	70	PROTECT
21	SS	71	RUN
22	ADDR DSBL	72	PRDY
23	DO DSBL	73	PINT
24	ø2	74	PHOLD
25	ø1	75	PRESET
26	PHLDA	76	PSYNC
27	PWAIT	77	PWR
28	PINTE	78	PDBIN
29	A5	79	A0
30	A4	80	A1
31	A3	81	A2
32	A15	82	A6
33	A12	83	A7
34	A9	84	A8
35	DO1	85	A13
36	DO0	86	A14
37	A10	87	A11
38	DO4	88	DO2
39	DO5	89	DO3
40	DO6	90	DO7
41	DI2	91	DI4
42	DI3	92	DI5
43	DI7	93	DI6
44	SM1	94	DI1
45	SOUT	95	DI0
46	SINP	96	SINTA
47	SMEMR	97	SWO
48	SHLTA	98	SSTACK
49	CLOCK (2MHz)	99	POC
50	GND	100	GND

The S-100 Bus Structure

PROGRAM LISTING

```
8400 =          BASE    EQU     8400H
                ;
                ;       EQUATES FØR LINKAGE TØ CP/M
                ;
9000 =          CURDSK  EQU     110000Q
9001 =          TRACK   EQU     110001Q
9002 =          SECTØR  EQU     110002Q
9003 =          CHARAD  EQU     110003Q
9005 =          BUFADR  EQU     110005Q
                ;
                ;       INTERNAL EQUATES FØR DISK PRIMITIVE CØDE
                ;
9007 =          FSECT   EQU     110007Q ;FIRST SECTØR TØ READ/WRITE
9008 =          LSECT   EQU     110010Q ;LAST SECTØR TØ READ/WRITE
9009 =          FDBUF   EQU     110011Q ;CURRENT TRACK BUFFER PØINTER
900B =          PUTSW   EQU     110013Q ;SECTØR WRITE SWITCH
900C =          CURSEC  EQU     110014Q ;CURRENT SECTØR ADDRESS
900E =          CHARCT  EQU     110016Q ;PRINT LINE LENGTH
900F =          WKBUF   EQU     110017Q ;TEMPØRARY BØØT WØRK AREA
                ;
```

```
                ;       CUSTØMIZED BASIC INPUT ØUTPUT SYSTEM LINKAGE
                ;
8400                    ØRG     BASE
8400 C33084     START:  JMP     HBØØT
8403 C33E84     LNKTBL: JMP     CBØØT
8406 C37184             JMP     VBØØT
8409 C30D85             JMP     CØNST
840C C31885             JMP     CØNIN
840F C32485             JMP     CØNØUT
8412 C3BE85             JMP     LIST
8415 C3E085             JMP     PUNCH
8418 C31885             JMP     READER
841B C30486             JMP     HØME
841E C33686             JMP     SELDSK
8421 C38186             JMP     SETTRK
8424 C3B686             JMP     SETSEC
8427 C3BB86             JMP     SETDMA
842A C3C186             JMP     READ
842D C30187             JMP     WRITE
                ;
                ;       SYSTEM BØØTSTRAP RØUTINE
                ;
8430 AF         HBØØT:  XRA     A
8431 320E90             STA     CHARCT
8434 320B90             STA     PUTSW
8437 320090             STA     CURDSK
843A 2F                 CMA
843B 320190             STA     TRACK
843E 310094     CBØØT:  LXI     SP,TØP
8441 0E0B               MVI     C,13Q
8443 CD2485             CALL    CØNØUT
8446 3E00               MVI     A,0
8448 D3C8               ØUT     310Q
844A 3E01               MVI     A,1
844C D3E4               ØUT     344Q
844E CDEB85             CALL    MØUT
8451 4D4F554E54         DB      'MØUNT CP/M DISK',EØT
8461 DBE0       BØØT1:  IN      340Q
8463 E680               ANI     200Q
8465 C26184             JNZ     BØØT1
8468 210000             LXI     H,0
846B 2B         BØØT2:  DCX     H
846C 7C                 MØV     A,H
846D B5                 ØRA     L
846E C26B84             JNZ     BØØT2
                ;
                ;       WARM BØØT RØUTINE
                ;
8471 310094     VBØØT:  LXI     SP,TØP
8474 CD0486             CALL    HØME
8477 010F90             LXI     B,WKBUF
847A CDBB86             CALL    SETDMA
847D 0E02               MVI     C,2
847F CDB686             CALL    SETSEC
8482 CDC186             CALL    READ
8485 A7                 ANA     A
8486 C2FB84             JNZ     BØØTFL
8489 3A1190             LDA     WKBUF+2
848C E6FC               ANI     374Q
848E FE6C               CPI     6CH     ;CHECK SIZE
8490 D2FB84             JNC     BØØTFL
8493 3C                 INR     A
8494 67                 MØV     H,A
8495 2E00               MVI     L,0
8497 E5                 PUSH    H
8498 44                 MØV     B,H
8499 4D                 MØV     C,L
849A CDBB86             CALL    SETDMA
849D 062E               MVI     B,46
849F C5         VBØØT1: PUSH    B
84A0 CDC186             CALL    READ
84A3 A7                 ANA     A
84A4 C2FB84             JNZ     BØØTFL
84A7 2A0590             LHLD    BUFADR
84AA 118000             LXI     D,128
84AD 19                 DAD     D
84AE 220590             SHLD    BUFADR
84B1 3A0290             LDA     SECTØR
84B4 3C                 INR     A
84B5 FE1B               CPI     27
84B7 DABC84             JC      VBØØT2
84BA 3E01               MVI     A,1
84BC 320290     VBØØT2: STA     SECTØR
84BF F5                 PUSH    PSW
84C0 4F                 MØV     C,A
84C1 CDB686             CALL    SETSEC
84C4 F1                 PØP     PSW
84C5 3D                 DCR     A
84C6 C2CE84             JNZ     VBØØT3
84C9 3C                 INR     A
84CA 4F                 MØV     C,A
84CB CD8186             CALL    SETTRK
84CE C1         VBØØT3: PØP     B
84CF 05                 DCR     B
84D0 C29F84             JNZ     VBØØT1
84D3 E1                 PØP     H
84D4 E5                 PUSH    H
84D5 110609             LXI     D,0906H
84D8 19                 DAD     D
84D9 220600             SHLD    6
84DC 11FD0B             LXI     D,0BFDH
84DF 19                 DAD     D
84E0 220100             SHLD    1
84E3 3EC3               MVI     A,303Q
84E5 320000             STA     0
84E8 320500             STA     5
84EB 2E00               MVI     L,0
84ED 062D               MVI     B,45
84EF 110384             LXI     D,LNKTBL
84F2 1A         VBØØT4: LDAX    D
84F3 77                 MØV     M,A
84F4 13                 INX     D
84F5 23                 INX     H
84F6 05                 DCR     B
84F7 C2F284             JNZ     VBØØT4
84FA C9                 RET
                ;
                ;       BØØTSTRAP ERRØR
                ;
84FB CDEB85     BØØTFL: CALL    MØUT
84FE 0D0A424F4F         DB      CR,LF,'BØØT FAILED',EØT
850C 76                 HLT
                ;
                ;       CØNSØLE STATUS RØUTINE
                ;
850D DB00       CØNST:  IN      0
850F E601               ANI     1
8511 C21685             JNZ     CØNST1
8514 2F                 CMA
8515 C9                 RET
8516 AF         CØNST1: XRA     A
8517 C9                 RET
                ;
                ;       CØNSØLE INPUT RØUTINE
                ;
8518 DB00       CØNIN:  IN      0
851A E601               ANI     1
851C C21885             JNZ     CØNIN
851F DB01               IN      1
8521 E67F               ANI     177Q
8523 C9                 RET
                ;
                ;       CØNSØLE ØUTPUT RØUTINE (VIDEØ DRIVER)
                ;
8524 79         CØNØUT: MØV     A,C
8525 FE0D               CPI     15Q     ;CARRIAGE RETURN
8527 CA4D85             JZ      CØNX1
852A FE0A               CPI     12Q     ;LINE FEED
852C CA6085             JZ      CØNX2
852F FE0B               CPI     13Q     ;VERTICAL TAB (HØME UP & CLEAR)
8531 CAA485             JZ      CØNX3
8534 2A0390             LHLD    CHARAD
8537 77                 MØV     M,A
8538 23                 INX     H
8539 7D                 MØV     A,L
853A E63F               ANI     77Q
853C FE3F               CPI     77Q
853E C24785             JNZ     CØNX0
8541 CD4D85             CALL    CØNX1
8544 C36085             JMP     CØNX2
8547 220390     CØNX0:  SHLD    CHARAD
854A 36A0               MVI     M,240Q
854C C9                 RET
854D 2A0390     CØNX1:  LHLD    CHARAD
8550 7E                 MØV     A,M
8551 E67F               ANI     177Q
8553 77                 MØV     M,A
8554 7D                 MØV     A,L
8555 E6C0               ANI     300Q
8557 6F                 MØV     L,A
8558 7E                 MØV     A,M
8559 F680               ØRI     200Q
855B 77                 MØV     M,A
855C 220390             SHLD    CHARAD
855F C9                 RET
8560 2A0390     CØNX2:  LHLD    CHARAD
8563 7E                 MØV     A,M
8564 E67F               ANI     177Q
8566 77                 MØV     M,A
8567 114000             LXI     D,64
856A 19                 DAD     D
856B 3ED0               MVI     A,320Q
856D BC                 CMP     H
856E C29C85             JNZ     CØNX2A
8571 7D                 MØV     A,L
8572 E63F               ANI     77Q
8574 5F                 MØV     E,A
8575 2100CF             LXI     H,147700Q
8578 19                 DAD     D
8579 220390             SHLD    CHARAD
857C 2100CC             LXI     H,146000Q
857F 01C003             LXI     B,960
8582 1140CC             LXI     D,146100Q
8585 1A         CØNX2B: LDAX    D
8586 77                 MØV     M,A
8587 23                 INX     H
8588 13                 INX     D
8589 0B                 DCX     B
858A 78                 MØV     A,B
858B B1                 ØRA     C
858C C28585             JNZ     CØNX2B
858F 0640               MVI     B,64
8591 3E20               MVI     A,' '
8593 77         CØNX2C: MØV     M,A
8594 23                 INX     H
8595 05                 DCR     B
8596 C29385             JNZ     CØNX2C
8599 2A0390             LHLD    CHARAD
859C 7E         CØNX2A: MØV     A,M
859D F680               ØRI     200Q
859F 77                 MØV     M,A
85A0 220390             SHLD    CHARAD
85A3 C9                 RET
85A4 2100CC     CØNX3:  LXI     H,146000Q
85A7 220390             SHLD    CHARAD
85AA 010004             LXI     B,1024
85AD 3620       CØNX3A: MVI     M,' '
85AF 23                 INX     H
85B0 0B                 DCX     B
85B1 78                 MØV     A,B
85B2 B1                 ØRA     C
85B3 C2AD85             JNZ     CØNX3A
85B6 2A0390             LHLD    CHARAD
85B9 7E                 MØV     A,M
85BA F680               ØRI     200Q
85BC 77                 MØV     M,A
85BD C9                 RET
                ;
                ;       LIST ØN THE TELETYPE
                ;
85BE 79         LIST:   MØV     A,C
85BF FE0D               CPI     CR
85C1 CADC85             JZ      LIST1
85C4 FE0A               CPI     LF
85C6 CAE085             JZ      LIST2
85C9 FE20               CPI     40Q
85CB D8                 RC
85CC FE60               CPI     140Q
85CE D0                 RNC
85CF 3A0E90             LDA     CHARCT
85D2 FE48               CPI     72
85D4 C8                 RZ
85D5 3C                 INR     A
85D6 320E90             STA     CHARCT
85D9 C3E085             JMP     LIST2
85DC AF         LIST1:  XRA     A
85DD 320E90             STA     CHARCT
85E0 DB00       LIST2:  IN      0
85E2 E680               ANI     200Q
85E4 C2E085             JNZ     LIST2
85E7 79                 MØV     A,C
85E8 D301               ØUT     1
85EA C9                 RET
                ;
                ;       PUNCH RØUTINE
                ;
85E0 =          PUNCH   EQU     LIST2
                ;
                ;       READER RØUTINE
                ;
8518 =          READER  EQU     CØNIN
                ;
                ;       MESSAGE ØUTPUT RØUTINE
                ;
85EB E3D5C5F5   MØUT:   XTHL! PUSH D! PUSH B! PUSH PSW
85EF 7E                 MØV     A,M
85F0 23                 INX     H
85F1 FE04               CPI     EØT
85F3 CAFF85             JZ      MØUT1
85F6 E5                 PUSH    H
85F7 4F                 MØV     C,A
85F8 CD2485             CALL    CØNØUT
```

```
85FB E1                 PØP     H
85FC C3EF85             JMP     MØUT+4
85FF F1C1D1E3  MØUT1:   PØP PSW! PØP B! PØP D! XTHL
8603 C9                 RET
               ;********************
               ;
               ;        ENTER MACRØ FØR DISK INPUT/ØUTPUT HERE
               ;
               ;        TRK1Ø=TRACK BUFFER
               ;
               ;        SECT1Ø=NØ BUFFER
               ;
               ;********************
               ;
               ;        FLØPPY DISK PRIMITIVE CØDE
               ;
               ;        ENTRY PØINTS:
               ;
               ;                HØME    - RESTØRE THE DISK
               ;
               ;                SELDSK  - SELECT A DRIVE
               ;
               ;                SETTRK  - SETS TRACK #
               ;
               ;                SETSEC  - SETS SECTØR #
               ;
               ;                SETDMA  - SETS DISK BUFFER ADDRESS
               ;
               ;                READ    - READ A SECTØR
               ;
               ;                WRITE   - WRITES A SECTØR
               ;
               ;
               ;        RESTØRE RØUTINE
               ;
8604 3E03      HØME:    MVI     A,3
8606 D3E4               ØUT     344Q
8608 211186             LXI     H,HØME1
860B 3E0F               MVI     A,17Q
860D D3E0               ØUT     340Q
860F FB                 EI
8610 76                 HLT
8611 3E01      HØME1:   MVI     A,1
8613 D3E4               ØUT     344Q
8615 DBE0               IN      340Q
8617 E698               ANI     230Q
8619 CA3086             JZ      HØME2
861C CDEB85             CALL    MØUT
861F 0D0A524553         DB      CR,LF,'RESTØRE ERRØR',EØT
862F 76                 HLT
8630 0E00      HØME2:   MVI     C,0
8632 CD8186             CALL    SETTRK
8635 C9                 RET
               ;
               ;        SELECT THE CURRENT DISK
               ;
8636 210090    SELDSK:  LXI     H,CURDSK
8639 79                 MØV     A,C
863A BE                 CMP     M
863B C8                 RZ
863C 77                 MØV     M,A
863D C641               ADI     'A'
863F F5                 PUSH    PSW
8640 CDEB85             CALL    MØUT
8643 0D0A4D4F55         DB      CR,LF,'MØUNT DISK ',EØT
8651 F1                 PØP     PSW
8652 4F                 MØV     C,A
8653 CD2485             CALL    CØNØUT
8656 CDEB85             CALL    MØUT
8659 2026205459         DB      ' & TYPE RETURN',EØT
8668 CD1885    SELD3:   CALL    CØNIN
866B FE0D               CPI     CR
866D C26886             JNZ     SELD3
8670 4F                 MØV     C,A
8671 CD2485             CALL    CØNØUT
8674 0E0A               MVI     C,LF
8676 CD2485             CALL    CØNØUT
8679 DBE0      SELD1:   IN      340Q
867B E680               ANI     200Q
867D C27986             JNZ     SELD1
8680 C9                 RET
               ;
               ;        SET TRACK RØUTINE
               ;
8681 3E03      SETTRK:  MVI     A,3
8683 D3E4               ØUT     344Q
8685 210190             LXI     H,TRACK
8688 71                 MØV     M,C
8689 DBE1               IN      341Q
868B BE                 CMP     M
868C CAB186             JZ      STRK2
868F 7E                 MØV     A,M
8690 D3E3               ØUT     343Q
8692 3E1F               MVI     A,37Q
8694 219B86             LXI     H,STRK1
8697 D3E0               ØUT     340Q
8699 FB                 EI
869A 76                 HLT
869B DBE0      STRK1:   IN      340Q
869D E698               ANI     230Q
869F CAB186             JZ      STRK2
86A2 CDEB85             CALL    MØUT
86A5 5345454B20         DB      'SEEK ERRØR',EØT
86B0 76                 HLT
86B1 3E01      STRK2:   MVI     A,1
86B3 D3E4               ØUT     344Q
86B5 C9                 RET
               ;
               ;        SET SECTØR RØUTINE
               ;
86B6 210290    SETSEC:  LXI     H,SECTØR
86B9 71                 MØV     M,C
86BA C9                 RET
               ;
               ;        SET DMA LØCATIØN
               ;
86BB 69        SETDMA:  MØV     L,C
86BC 60                 MØV     H,B
86BD 220590             SHLD    BUFADR
86C0 C9                 RET
               ;
               ;        READ A SECTØR
               ;
86C1 060A      READ:    MVI     B,10
86C3 3E03      READ1:   MVI     A,3
86C5 D3E4               ØUT     344Q
86C7 3A0290             LDA     SECTØR
86CA D3E2               ØUT     342Q
86CC 2A0590             LHLD    BUFADR
86CF EB                 XCHG
86D0 21ED86             LXI     H,READ3
86D3 3ED0               MVI     A,320Q
86D5 D3E0               ØUT     340Q
86D7 DBE0               IN      340Q
86D9 E620               ANI     40Q
86DB 3E88               MVI     A,210Q
86DD C2E286             JNZ     RHLD
86E0 3E8C               MVI     A,214Q
86E2 D3E0      RHLD:    ØUT     340Q
86E4 FB        READ2:   EI
86E5 76                 HLT
86E6 DBE3               IN      343Q
86E8 12                 STAX    D
86E9 13                 INX     D
86EA C3E486             JMP     READ2
86ED 3E01      READ3:   MVI     A,1
86EF D3E4               ØUT     344Q
86F1 DBE0               IN      340Q
86F3 E69C               ANI     234Q
86F5 CAFF86             JZ      READ4
86F8 05                 DCR     B
86F9 C2C386             JNZ     READ1
86FC 3E01               MVI     A,1
86FE C9                 RET
86FF AF        READ4:   XRA     A
8700 C9                 RET
               ;
               ;        WRITE A SECTØR
               ;
8701 060A      WRITE:   MVI     B,10
8703 3E03      WRITE1:  MVI     A,3
8705 D3E4               ØUT     344Q
8707 3A0290             LDA     SECTØR
870A D3E2               ØUT     342Q
870C 2A0590             LHLD    BUFADR
870F EB                 XCHG
8710 212D87             LXI     H,WRITE3
8713 3ED0               MVI     A,320Q
8715 D3E0               ØUT     340Q
8717 DBE0               IN      340Q
8719 E620               ANI     40Q
871B 3EA8               MVI     A,250Q
871D C22287             JNZ     WHLD
8720 3EAC               MVI     A,254Q
8722 D3E0      WHLD:    ØUT     340Q
8724 FB        WRITE2:  EI
8725 76                 HLT
8726 1A                 LDAX    D
8727 D3E3 .             ØUT     343Q
8729 13                 INX     D
872A C32487             JMP     WRITE2
872D 3E01      WRITE3:  MVI     A,1
872F D3E4               ØUT     344Q
8731 DBE0               IN      340Q
8733 E6FC               ANI     374Q
8735 CA3F87             JZ      WRITE4
8738 05                 DCR     B
8739 C20387             JNZ     WRITE1
873C 3E01               MVI     A,1
873E C9                 RET
873F AF        WRITE4:  XRA     A
8740 C9                 RET
               ;
               ;        EQUATE LINKAGE TØ FDØS
               ;
0001 =         RDCØN    EQU     1Q      ;READ CHARACTER FRØM CØNSØLE
0002 =         WRCØN    EQU     2Q      ;WRITE CHARACTER TØ CØNSØLE
000F =         ØPENFL   EQU     17Q     ;ØPEN A FILE
0010 =         CLØSFL   EQU     20Q     ;CLØSE A FILE
0016 =         MAKEFL   EQU     26Q     ;MAKE A NEW DIRECTØRY ENTRY
0015 =         PUTFL    EQU     25Q     ;PUT A SECTØR TØ DISK
001A =         SDMAFL   EQU     32Q     ;SET DISK DMA ADDRESS
0004 =         EØT      EQU     4Q      ;END ØF TRANSMISSIØN
000A =         LF       EQU     12Q     ;LINE FEED
000B =         VT       EQU     13Q     ;VERTICAL TAB (HØME-UP & CLEAR)
000D =         CR       EQU     15Q     ;CARRIER RETURN
001B =         ESC      EQU     33Q     ;ESCAPE CØDE
0000 =         BØØT     EQU     0000H   ;ENTRY TØ BØØTSTRAP
0005 =         ENTRY    EQU     0005H   ;ENTRY ADDRESS ØF FDØS
005C =         TFCB1    EQU     005CH   ;FIRST DEFAULT FILE CØNTRØL BLØC
006C =         TFCB2    EQU     006CH   ;SECØND FILE NAME
0080 =         TBUFF    EQU     0080H   ;DEFAULT BUFFER ADDRESS
0100 =         TBASE    EQU     0100H   ;BASE ØF TRANSIENT AREA
9400 =         TØP      EQU     112000Q ;TØP ØF STACK
8741                    END     START
```

REFERENCES

Advanced Micro Devices, Inc., *The 8080A/9080A MOS Microprocessor Handbook,* Sunnyvale, CA, March 1977.
Advanced Micro Devices, Inc., 915 DeGuigne Drive, Sunnyvale, CA 94086

Intel Corporation, *Intel 8080 Microcomputer Systems User's Manual,* Santa Clara, CA, September 1975.
Intel Corporation, 3065 Bowers Ave., Santa Clara, CA 95051

Western Digital Corporation, *FD1771 Floppy Disk Controller Application Note,* Newport Beach, CA, December 1976.
Western Digital Corporation, 3128 Red Hill Ave., Box 2180, Newport Beach, CA 92663

Western Digital Corporation, *FD1771 Floppy Disk Formatter/Controller Data Sheet,* Newport Beach, CA, December 1976.
Western Digital Corporation, 3128 Red Hill Avenue, Box 2180, Newport Beach, CA 92663

Western Digital Corporation, *FD1771A/B-01 Floppy Disk Formatter/Controller Data Sheet.* Newport Beach, CA, March 1977.
Western Digital Corporation, 3128 Red Hill Ave., Box 2180, Newport Beach, CA 92663

National Semiconductor Corporation, *INS1771-1 Floppy Disk Formatter/Controller Data Sheet,* Santa Clara, CA, April 1977.
National Semiconductor Corporation, 2900 Semiconductor Drive, Santa Clara, CA 95051.

INTERFACE AGE
BACK ISSUES

Available in Limited Quantities

Vol. 1, Issue 5, APRIL, 1976
Introduction to Microprocessor Technology; Bubble Memories Are Coming; Calculatin' Engines; Teleprinter Maintenance; Altair Alterations.

Vol. 1, Issue 11, OCTOBER, 1976
National's New Portable Terminal; SA-400 Mini-floppy; CSC—Experimentor 300/600; Software Power for Your 6800; ESP-1 Software Package

Vol. 1, Issue 12, NOVEMBER, 1976
Build a Simple A to D; Super Chip FD1771; RCA ASCII Keyboard Modifications; Protecting Stored Programs; New Product Guide; SWTPC 6800 ROBIT-1

Vol. 2, Issue 2, JANUARY, 1977
A CRT Terminal Using the M6800 Chip Family; AMI's EVK Series Microcomputer; Build a 3-Digit A/D; Microcomputer Design Aids; Cromemco Bytesaver™

Vol. 2, Issue 3, FEBRUARY, 1977
8080 Octal Monitor Program; Legion: An Experiment in Artificial Intelligence; Microcomputer Stock Options; Building a 12-Bit A to D Converter

Vol. 2, Issue 4, MARCH, 1977
Menace of the Micro World; New Product Guide; The Qube; Card of the Month—Cromemco T.V. Dazzler™; Z-80 MITS 12K Extended BASIC Patches.

Vol. 2, Issue 6, MAY, 1977
Computrac 2000; The Floppy-ROM™ Experiment; Robert Uiterwyk's 4K BASIC Interpreter Program; Help Your Computer Understand Your Voice

Vol. 2, Issue 8, JULY, 1977
Diablo Output Driver Routine; Some Further Notes on Robert Uiterwyk's Floppy ROM 4K BASIC; Microcomputerized Combination Lock; PIA Test-IOTST

Vol. 2, Issue 9, AUGUST, 1977
The Shadow of What?; A KIM-1 Sidereal/Solar Clock; Solar Eclipse Prediction by Microcomputer; Viking UPLINK/DOWNLINK; Star-Ship Simulation-Part I

Vol. 2, Issue 10, SEPTEMBER, 1977
General Ledger Program; Microcomputers: The Intelligent Terminals; Star-Ship Simulation-Part II; PerSci Intelligent Floppy Disc Controller

Vol. 2, Issue 11, OCTOBER, 1977
Computers for Survival; Portable Automated Mesonet; The Energy/Environment Simulators; Biorhythm; Assembly Language Structured Programming; Video Chase for 8080/VDM

Vol. 2, Issue 12, NOVEMBER, 1977
Point Humans: Optical Perception of People and Computers; A Byte of Music; General Ledger Package: The Micro Bookmaker, Conclusion; Number Base Conversion Program

Vol. 3, Issue 1, JANUARY, 1978
The Word Processor; Medical Accounts Receivable Package; How to Buy an Apartment House; Turning Toward Meca; Computer(ese) Speech Writer; Tax Calculation Program

Vol. 3, Issue 2, FEBRUARY, 1978
Wallet-size Spaceship Simulator — Part I; LSI-11 Microcomputers in Hospital ICU's; Structured Programming in BASIC; Electron Beam Pattern Generator; World Power — Game

INTERFACE AGE Magazine Dept. BI · P.O. Box 1234, Cerritos, CA 90701

Name (Print) ______________ Address ______________ City ______________ State ________ Zip ________

Please send me:

Issue	Qty	Price	Total	Issue	Qty	Price	Total	Issue	Qty	Price	Total
APRIL 1976		2.25*		MARCH 1977		2.25*		OCTOBER 1977		2.50*	
OCTOBER 1976		2.25*		MAY 1977		2.25*		NOVEMBER 1977		2.50*	
NOVEMBER 1976		2.25*		JULY 1977		2.50*		JANUARY 1978		2.50*	
JANUARY 1977		2.25*		AUGUST 1977		2.50*		FEBRUARY 1978		2.50*	
FEBRUARY 1977		2.25*		SEPTEMBER 1977		2.50*					

*Price includes 50¢ for postage and handling.

TOTAL ENCLOSED $ ______________

VISA® # ______________ master charge ______________ Exp. Date ________ Sig. ______________

You may photocopy this page if you wish to keep your INTERFACE AGE intact. Please allow six weeks for delivery.

Computer Tutorial — Part IV

More Memories

By Roger H. Edelson, Hardware Editor

This is the third article in this series to discuss memories. The heavy emphasis placed on memory is due to the major role this subsystem plays in the small computer system, both from a cost and size viewpoint. Until the advent of the semiconductor memory, the small computer user had no really viable means of providing the short term memory function required by almost any computer application. As mentioned in an earlier article, numerous schemes were tried including used core planes, tape loops, delay-lines, etc. The problem with all these attempts was that they were not available to a large proportion of the possible users, and they were only suitable for the more knowledgeable tinkerers.

Speaking of delay lines, I just came across a 1968 model desk calculator that uses a magnetostrictive delay line for its entire memory. The line stores 408 bits organized as 5 registers of 68 bits each and 1 register of 68 bits. The single register portion of the line circulates five times as fast as the five-register segment, making it more available to the calculation logic. This is achieved on a single line by intermixing the single register with the storage of the five registers as shown in Figure 1.

This is a unique method of solving the memory problem; unfortunately the line is packaged in a box 7x4x1½ inches. The electronics which includes two write amplifiers, a read amplifier and address decoding, takes up a full card. With the presently available semiconductor memories, this storage function could be implemented with about three chips. You can see why semiconductor memories have replaced almost all other competing devices for small-to-medium size fast-access applications.

The first semiconductor memories were bipolar devices fabricated using S-R (Set-Reset) flip-flops with the addition of addressing structures. The basic memory cell of such a device is shown in Figure 2. As can be seen, cross-coupled NOR gates are used to form the flip-flop, which is only accessible when the word line is high. The output of the flip-flop is gated on to the "0" line for reading. An early organization of this type of memory resulted in a 64-bit RAM (AM29700) shown in Figure 3. $D_{0\text{-}3}$ are the data input lines and $O_{0\text{-}3}$ are the data output lines. WE controls the write/read operations and the CS line is used to provide for memory expansion. A device built like this, using Schottky technology is capable of a 17ns access time.

Sixty-four bits on a chip, while representing a significant advance over previous storage densities, still did not meet the needs of the microcomputer industry. At this point MOS technology began to flex its muscles with the development of static and dynamic memories. Both types of memory serve the same market and were developed at about the same time. The static devices have the edge in ease of use and speed, while the dynamic types lead in the power-cost function.

Figure 4 is a diagram of the basic static cell — a six-transistor flip-flop formed of depletion mode devices. Data is stored as charge on the gate of either Q_3 or Q_4, the cross-connected transistors. Devices Q_2 and Q_5 function as high-value load resistors. It is interesting to note that it is easier to fabricate resistors as degerate active devices (two terminal MOSFETs) than to add the extra process steps necessary to form a passive resistor. For this device, we will use the convention that if Q_3 is "ON" then a "0" is stored in the cell, and if Q_4 is "ON" then a "1" is stored. If Q_3 is "ON" it holds the gate voltage of Q_4 low enough to keep device Q_4 "OFF" and the cell will hold that state. Transistors Q_1 and Q_6 form switches which connect the cell to the data read/write lines when that particular row has been selected.

Column selection is done with the data lines. Figure 5 shows the organization of a typical 1024 words x 1 bit memory (Intel 2102A), and Figure 6 shows a more detailed view of the internal data path. Note how the column select line actually selects the write buffer and a read output gate.

For a read operation, a sense amplifier connected to both I/O "0" and I/O "1" outputs of each column detects the state of the selected storage cell inthat column. If Q_3 is on, then Q_4 is off, and when Q_6 is turned on (ROW SELECT) current will flow in the I/O "0" line. Current will flow in the opposite line (I/O "1") if a "1" is stored in the cell.

To set the state of the storage cell, a write buffer controls the level of the I/O lines. If it is desired to write a "1" then the I/O "1" line is taken to approximately V_{cc} while the opposite line is held low (V_{ss}). This action will

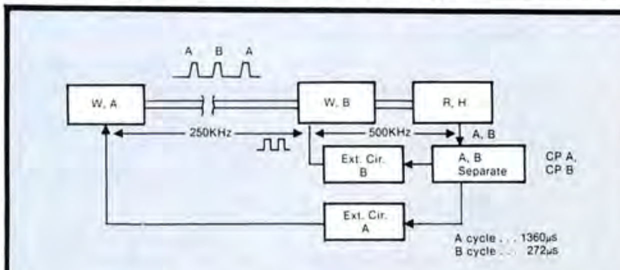


Figure 1. Delay Line Circuit.

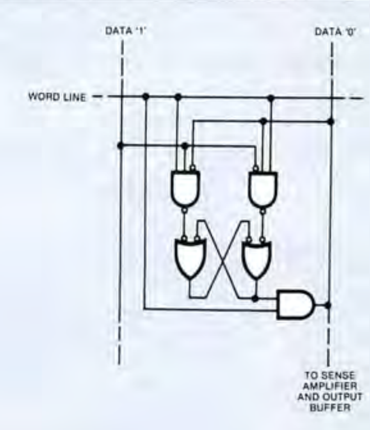


Figure 2. Basic Memory Cell.

override the previous state of the storage cell and the new information will be stored. A chip enable line is shown which gates both the internal data busses, allowing easy memory expansion.

A newer entry into the RAM field is the static memories utilizing the C-MOS (complementary-metal-oxide-semiconductor) process. One-kilobit devices are available with access time under 350ns and supply currents of less than 100 microamperes. In these memories, the load devices are replaced with P-channel devices. Static RAMs can be built as either synchronous or asynchronous devices. Asynchronous devices will allow an address change at any time with no loss of data, while synchronous memories must have a clock pulse to disable the chip during address changes. Figure 7 shows the block diagram of the Intersil IM6523, a 256 x 1 Asynchronous RAM.

The ability to store charge on a capacitor for an appreciable time is utilized in the MOS dynamic RAMs. The storage cell of a typical dynamic RAM (in this case the Intel 2107B) is shown in Figure 8 along with its associated Input/Output circuitry. This is a 4096 word x 1 bit memory organized using a 64 x 64 memory array. The overall memory organization is shown in Figure 9.

Operation of this memory is dependent upon proper design of the sense amplifier. This device not only senses the small voltage changes when the cell is read, but it then latches to the appropriate state to provide a logic level output and rewrite the data back into the cell. The data sense/latch (S/A) is shown in Figure 10. It should be noted that this memory is of the inverting type — that is, the data output at the output pin is the logical inverse of the data written into memory. The operation of the memory depends on the level of the voltage stored by capacitor C_{STG} — if it is above V' a logic "1" will be outputted, and if the stored voltage is below V' then a logic "0" will be read.

Consider first a read operation where the storage capacitor C_{STG} is discharged (node 1 is at V_{ss}). Before the chip enable line is brought high, the bit sense lines will be at a level of V' due to the on condition of transistors Q_1. When the address lines have stabilized, the proper column select line is brought high. This turns on device Q_2 and the storage capacitor is electrically connected to the bit sense line. The charge on the storage capacitor is then distributed between C_{STG} and $C_{I/O}$ (the parasitic capacitance of the bit sense line). Since the parasitic capacitor is much larger than the value of C_{STG} (also a parasitic capacitor), the voltage change of the bit line will be very small.

The sense/latch amplifier is a cross-coupled pair of transistors. The voltage level V' is such that both devices will be in their active regions, and the circuit will act as a very high gain regenerative amplifier. Therefore, if the left bit line is lower than the right bit line, Q_3 will conduct more heavily than Q_4. This operation is regenerative and very quickly the amplifier will latch-up with bit-sense-left going to V_{SS} and bit-sense-right going to nearly V_{DD} (V_X). Note that during the read operation the previous contents of the cell are destroyed and the read operation is effectively a destructive read. The action of the sense/latch amplifier is such that the information is automatically rewritten back into C_{STG} and therefore the process is transparent to the user, who in effect sees an NDRO device. A write operation is identical to the rewrite portion of the read cycle. In the case of a "write," the new information just overrides the state of the sense/latch and the desired logic level is written into the cell.

Since the information in a dynamic RAM is stored in a capacitor, it is subject to a voltage change because of leakage currents caused by parasitic resistors. For this reason the data must be constantly rewritten, or refresh-

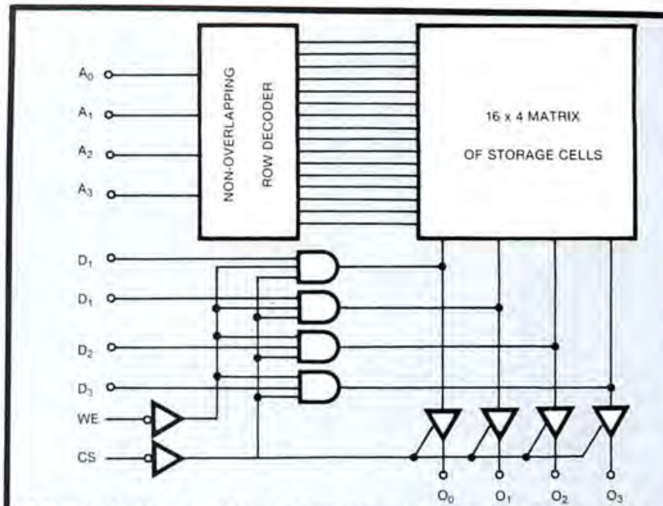


Figure 3. Logic Block Diagram for the AM29700, 64-bit.

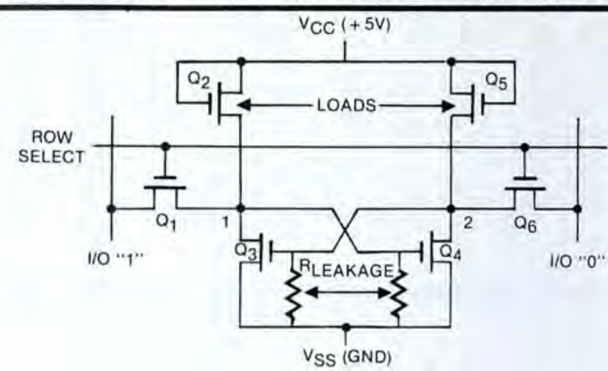


Figure 4. Basic MOS Static Memory Cell.

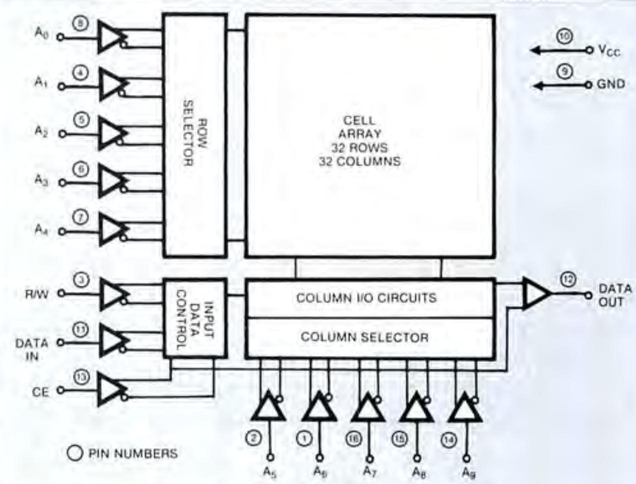


Figure 5. 2102 Block Diagram

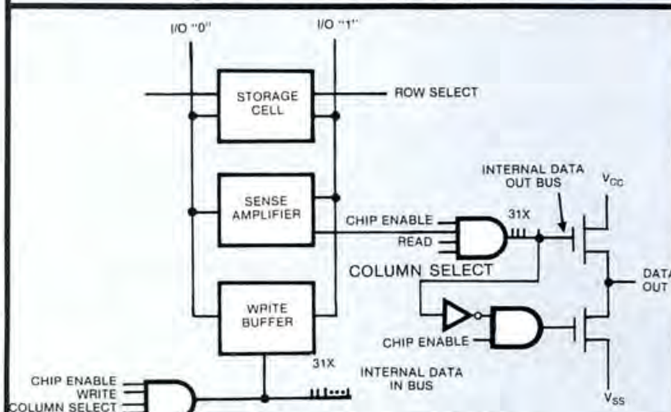


Figure 6. Internal Data Path.

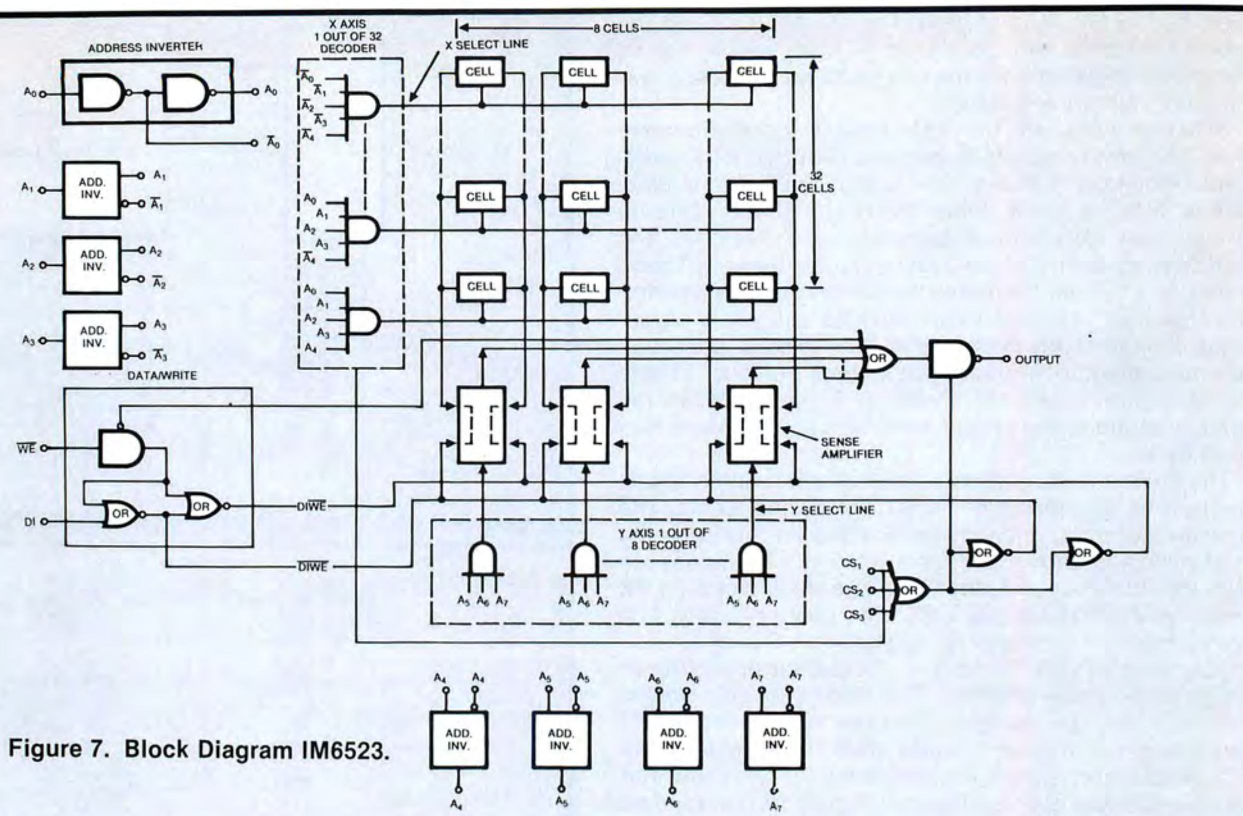


Figure 7. Block Diagram IM6523.

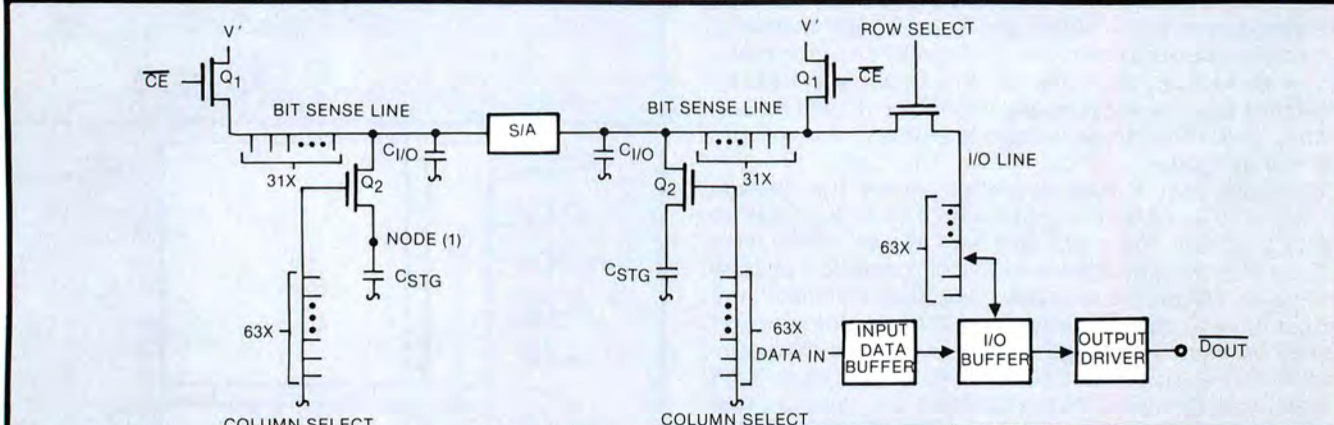


Figure 8. 2107B Memory Cell and Associated I/O Circuitry.

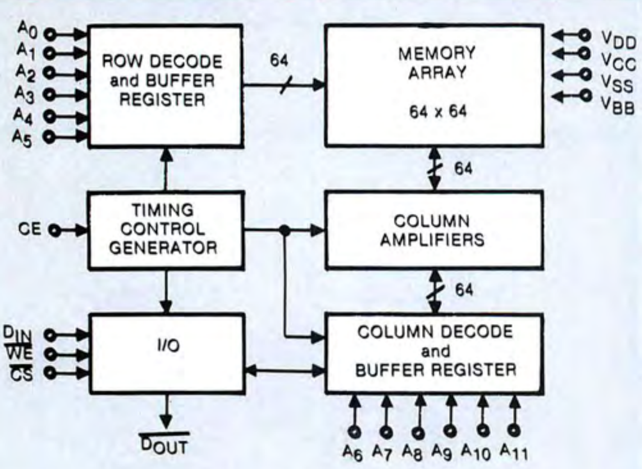


Figure 9. 2107B Block Diagram.

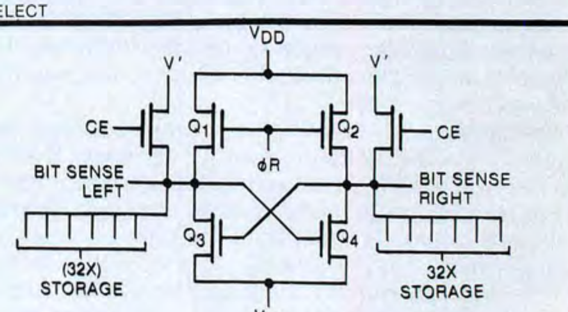


Figure 10. Data Sense/Latch Amplifier (S/A).

ed. This can be done by either a read cycle, a write cycle, or a read-modify write cycle. As each column has a sense/latch amplifier, the easiest way to accomplish this refresh is to perform one read cycle on each of the 64 row addresses. For the 2107B, the maximum refresh time is 2 ms. In earlier memory subsystem designs, the refresh cycle was done by halting the MPU. This would introduce delays into the computer operation, or would cause flashes or gaps in a video field. Later designs accomplish refresh during states of the MPU when memory access is not required — cycle stealing. The Z-80 provides a dedicated control to implement this transparent refresh.

The dynamic RAM provides high storage densities because of the small substrate area required by the storage cell. The storage capacitor is a parasitic element and therefore does not add to the real estate. The disadvantage is a somewhat higher subsystem complexity over static memories. The dynamic memory also provides a power saving over the standard static MOS memories. The C-MOS devices challenge the power dissipation superiority of the dynamic devices, but at a significantly higher cost. As the state of the art matures, various memories process improvements will provide increased storage densities with lower power and costs.

Figure 11 shows the organization of the Intel 2117 memory. This is a 16K word x 1 bit memory with 150 ns access time and a power dissipation of only 465mW. The same basic memory organization can be used to produce a read only memory (ROM) or a programmable read-only memory (PROM). Figure 12 shows the block diagram of the AMI 56834, a 512 x 8 bit uv erasable and electrically reprogrammable memory. This device provides for bulk erasure of the memory by exposing the chip to ultra-violet light through the transparent lid. After this bulk erasure, a new pattern may be stored.

Other types of PROMs can be built by using fusible Ni-Cr links. Once a fusible link PROM has been written, the pattern cannot be erased, although additional logic "1"'s can be placed over any logic "0".

Other types of memory organizations may be designed using semiconductor storage elements, among them FIFOs (First-In-First-Out) buffer registers, 16 bit (4 word x 4 bit) register files, and a 4 word x 2 bit content addressable memory. This later device provides the novel capability of data association, where the memory can be searched for a word to match a specific word. Actually matching may be done a bit basis, further expanding the applications may be done on a bit basis, further expanding the applications of this unique element. Figure 13 provides a logic diagram of this device, the Signetics 8220 CAM element.

Next month, we will cover the heavyweights of the non-moving memory field — the CCD's (charge coupled devices), and magnetic bubble memories.□

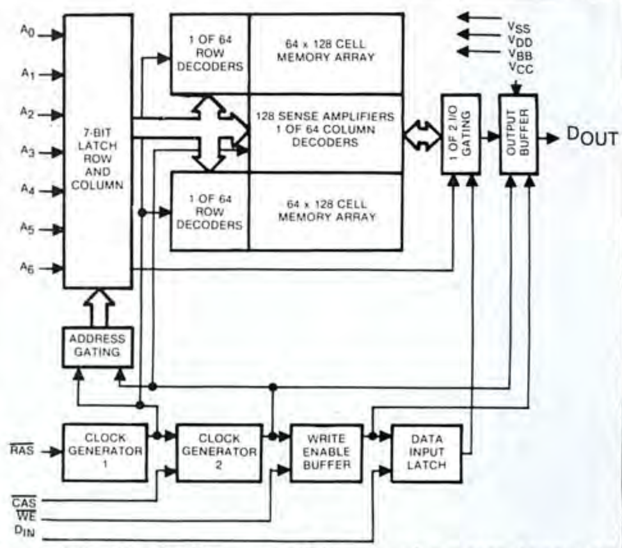


Figure 11. Block diagram of "second generation" 16K RAM. 2117 family.

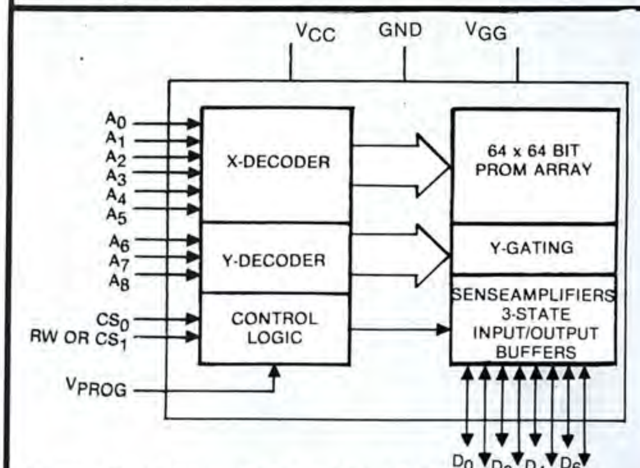


Figure 12. Block diagram AMI S6834.

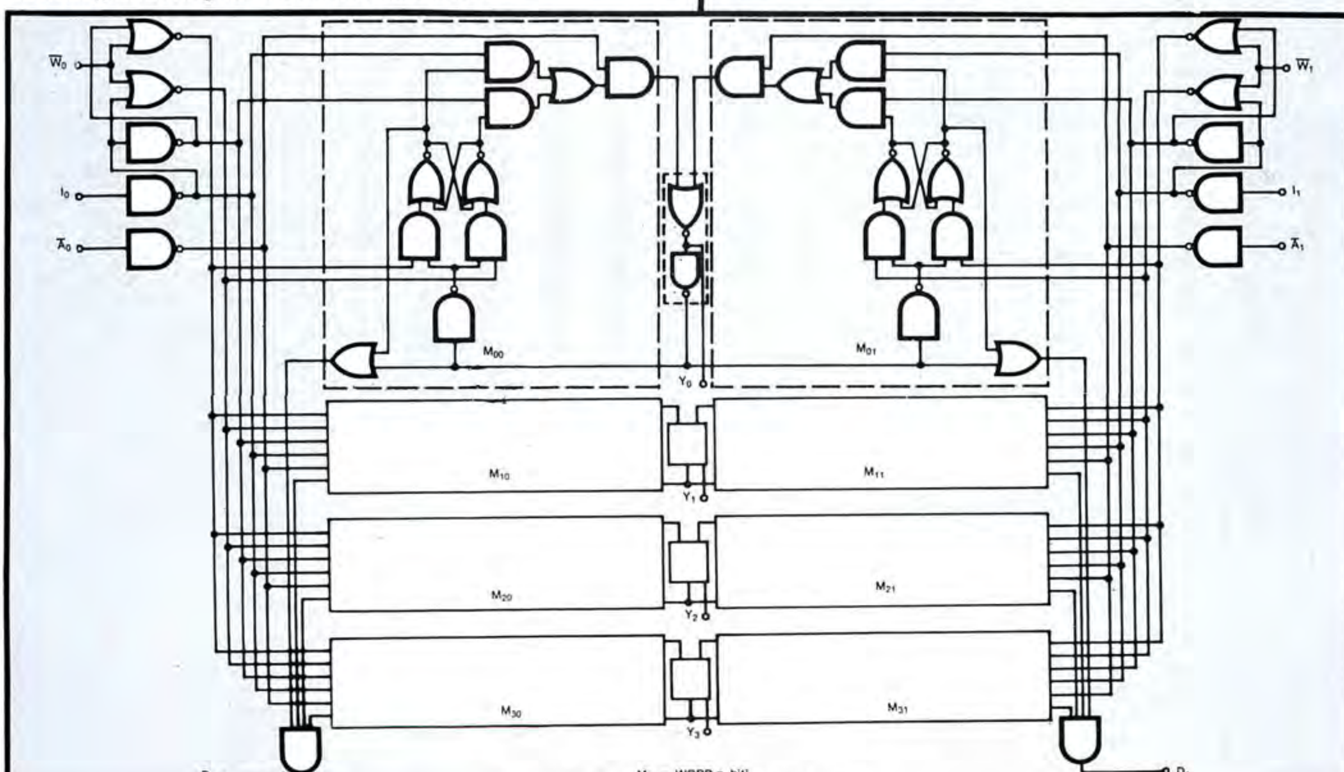


Figure 13. Signetics 8220 CAM.

BASIC's Token of Good Fortune

By Tim Ryan
University Data Systems

Most readers of hobby computing magazines have by now done some programming in BASIC. Any issue will contain a sample showing just what a program written in the BASIC language looks like. The more sophisticated reader may be wondering by now just how does the computer manage to keep all the statements and variables in a program straight in memory. Particularly, how does BASIC allow the flexibility of changing code so conveniently midstream during the interactive debugging stage of writing a program. Some readers may even be thinking of writing changes to BASIC to give them special features they need.

In order to reach the broad audience of users who have had experience programming in BASIC, but not in other languages, several concepts should be outlined. There are several levels of intricacy constantly occurring and invisible to you as a BASIC user. If you think of turning on a TV set and watching it, you do not have to be aware of the scanning of the beam across the screen and of the drift of the station frequency you have chosen. These are taken care of automatically by the electronics of the set. Likewise, if you are running BASIC, you are not concerned about the signals passing back and forth on the bus, or the translation of instructions by the microcode or logic of your processor. These are handled automatically. If you begin to think of your computer in terms of assembly language rather than just BASIC, there are still several other levels of subtlety that go unseen. These are located in your operating system and your specific flavor of BASIC.

OPERATING SYSTEM

You may be so used to working in BASIC that you don't think much about your operating system. It handles the routines kinds of chores that have to be done regardless of what language you are programming in. The bigger a computer is, the more diverse it must be in order to talk to terminals, printers, communication lines, disks, tapes, and other memory devices. If more than one person can run something on the computer at a time, it must schedule each job depending on its needs. Your operating system may be called the "executive" or the "monitor" but no matter what its name, it is the unseen workhorse that BASIC must communicate with to get certain things done. There are times when the computer may seem to idle but the monitor never sleeps. It "talks to itself" in a diddle pattern during which it is constantly checking (or "polling") to see if someone needs its help.

COMPILERS VS. INTERPRETERS

The operating system should not be fussy about who it's willing to help. It should be able to handle BASIC, FORTRAN, PL/I or any other language that you want to send it. There is nothing mysterious about each language, each is just another program that makes it easier for you to be understood by the computer and still be economical with your own time. One of these programs may translate a high level language into compiled assembly language, essentially casting the code in concrete. Once this is done with a language like FORTRAN, the compiler has done its job and from now on the resulting 'object' code produced by the compiler will function stand alone.

In contrast, BASIC is sometimes another kind of high level program, one called an interpreter. An interpreter translates the high level language it was designed for, statement by statement, symbol by symbol, each time the program is run, even if the statement is executed several times in a loop in the program. This has the advantage of being very flexible to allow for last minute debugging changes, but much slower than a compiled language where everything has been pre-translated. Another disadvantage is that the BASIC must always be present in memory to perform the translation. To combine the best of both worlds, some companies produce two versions of a language, one that runs interactively as an interpreter, and a second that can compile the resulting debugged code. This, of course, doubles their development and maintenance costs.

INTERPRETIVE COMPILER

In an effort to economize on speed and money, a better BASIC will combine the features of a compiler and interpreter to produce an intermediate form of code in memory. This code is more compact and will execute faster than the source code statements you type into the computer. It is not, however, as fast as compiled language and the BASIC must still be present in the memory at all times to translate this intermediate code.

200	\	231	STOP	262	=>	313	PI
201	FOR	232	END OF PROG	263	<>	314	SYS(
202	GOSUB	233	OPEN	264	><	315	RND(
203	GO TO	234	CLOSE	265	<	316	RND
204	ON	235	OVERLAY	266	>	317	SIN(
205	IF	236	CHAIN	267	=	320	COS(
206	INPUT	237	KILL	270	(	321	SQR(
207	LINPUT	240	NAME	271	"	322	ATN(
210	LET	241	&	272	'	323	EXP(
211	NEXT	242	^	273	.	324	LOG(
212	PRINT	243	*	274	#	325	LOG10(
213	RETURN	244	/	275	FN	326	ABS(
214	RESTORE	245	+	276	FOR OUTPUT	327	INT(
215	RESET	246	(UNARY)	277	AS FILE	330	SGN(
216	READ	247	-	300	FOR INPUT	331	BIN
217	CALL	250	(TERM)	301	%	332	OCT
220	IF END	251	,	302	$	333	LEN(
221	LET VF	252	)	303	DOUBLE BUF	334	ASC(
222	DIM	253	TO	304	RECORDSIZE	335	CHR$(
223	COMMON	254	STEP	305	FILESIZE	336	POS(
224	RANDOMIZE	255	THEN	306	MODE	337	SEG$(
225	REM	256	;	307	USING	340	VAL(
226	DEF	257	<=	310	LINE	341	TRM$(
227	DATA	260	=<	311	TAB	342	DAT$
230	END	261	>=	312	VF	343	STR$(

Table 1. Statement Tokens for Multi-user BASIC.

The process of compacting the code involved replacing each occurrence of a keyword like PRINT or INPUT in a program by a single character (or byte) called a token. There must be a unique token for each keyword (or verb). Unique tokens may also be set aside for arithmetic operation symbols like +, −, *, / and other special symbols like (,), ", ', and #. Table 1 shows the actual tokens used for one version of BASIC. The resulting compact code exists only in memory, although some versions of BASIC may allow you to store this 'core image' copy from memory onto disk or tape, to allow for a faster response time when a program is brought into memory.

There are also many other things happening during the compacting process that cause some of the memory that has been saved to be used for other purposes. Tables are created containing all line numbers and variables in the program and space is set aside for arrays that may appear in a DIM statement in the program. Once a program is in memory in this reduced form, BASIC must undo the compaction process every time a command like LIST is typed, to allow the program to be printed at a terminal, or when SAVE is typed to preserve the source code for the program on a device like a tape or disk.

Unfortunately, this process of compacting code is also called compiling, so as to confuse the distinction being made here between intermediate level of compilation like a BASIC interpretive compiler. The word compaction has been used temporarily in this section instead of compilation to keep the distinction clear.

A SPECIFIC EXAMPLE

It is not easy to choose a specific manufacturer's version of BASIC for discussion of details at the token level. As most of us are aware, BASIC comes in 57 varieties and 31 flavors. One of the primary advantages of a high level language is that it should be able to run on many different kinds of computers regardless of the manufacturer or hardware attached. Recently, a national minimal standard ANSI BASIC has been adopted. This will help create a minimum set of instructions that must function "the same," but it is unlikely to do away with the many unique features that each company might add above and beyond the standard. Since many of the hobby computers use the same microprocessor chips which have the identical instruction set, and since many hobby firms have only small, in house programming staffs, they are wary to supply listings of source code that would allow users to more easily understand or modify BASIC for a special purpose or enhancement.

One of the more sophisticated versions of BASIC that will run on the recently introduced Heath H-11, can also be purchased with source code. This code is available both on microfiche and on magnetic media to licensed users from the original manufacturer, Digital Equipment Corporation (DEC). DEC manufactures the LSI-11 CPU board used by Heath and furnishes the RT11 operating system along with BASIC-11.

BASIC-11 is really a family of languages meant to run on many different computers with many types of peripherals. The member chosen for this article is MU-BASIC, short for Multi-User BASIC. It needs the RT11 operating system and at least one disk or diskette drive to run. Since there is a large user's group called DECUS that Heath customers may join, they can take advantage of the experience of many sophisticated users who have paved the way by testing the software, lobbying for fancy special features and accumulating programs in the DECUS library.

BASIC	ASSEMBLY
Line number	Address (or Label)
Statement	Instruction
GO TO	Jump (JMP)
GOSUB	Jump to Subroutine (JSR)
RETURN	Return from Subroutine (RTS)
STOP	Halt (HALT)
LET A=0	Clear A (CLR A)
LET B=A	Move A to B (MOV A,B)
LET A=A+1	Increment A (INC A)
LET A=A-1	Decrement A (DEC A)
LET B=A+B	Add A to B (ADD A,B)
LET B=B-A	Subtract A from B (SUB A,B)
IF A=B GO TO	Compare (CMP A,B) & Conditional Branch (BEQ)

Branching in an assembly language program can be to an absolute address but the goal is usually to use a label for a destination in a program rather than absolute addresses so that the program can be loaded into memory at varying places and still be able to run. The resulting code is called relocatable or position independent code (PIC)

Table 2. BASIC vs. Assembly Language list of analogous concepts.

MU-BASIC

MU-BASIC allows several different users to be running programs simultaneously. There are not many versions of BASIC that can run on hobby machines that are this powerful. Figure 1, which shows how memory is used, helps you to visualize where BASIC is and how a BASIC program is divided up.

The left portion of Figure 1 shows how the available memory is allocated or 'mapped.' This overall breakdown shows an area called locore that is set aside to furnish the current address of devices, as well as to give addresses to branch to if certain kinds of errors are discovered or 'trapped' during the execution of the program.

The highest address range (141362-157777) is set aside for the operating system. As mentioned earlier, an interpretive language must always be present in memory to translate the BASIC code. The MU-BASIC interpretive compiler is located between locore and program one in memory. Only one copy is needed to handle several different programs written in BASIC simultaneously. The remaining room is divided between the current number of BASIC programs being run. Each BASIC program or job has a fixed amount of memory allotted to it. Jobs do not pool available space to be alternately used by one job, then the other.

A SINGLE BASIC PROGRAM

The right half of Figure 1 is the show and tell guide to the organization of a typical program written in BASIC as it appears in memory. The space shown is exclusively for the use of one program. The upper and lower boundaries of the job space are fixed during a dialogue when MUBAS is started. Typing the command LENGTH will tell the number of words used and the number of words

```
 Addresses           Total Memory          Single Program
Decimal  Octal
                     ___________           _____________
00000    000000     :           :         :             :
                    :  LO CORE  :         :    HEADER   :
00319    000477     ~~~~~~~~~~~~~         :_____________:
                    ~~~~~~~~~~~~~         :             :
00680    001250     :           :         :             :
                    :  MU-BASIC :         :  USER CODE  :
25268    061264     :___________:         :_____________:
25269    061265     :           :         :             :
                    :   BASIC   :         :             :
                    : PROGRAM 1 :         : SYMBOL TABLE:
36863    107777     :___________:         :_____________:
36864    110000     :           :         :             :
                    :   BASIC   :         :             :
                    : PROGRAM 2 :         :   STRINGS   :
49905    141361     :___________:         :_____________:
49906    141362     :           :         :             :
                    :   RT11    :         :   ARRAYS    :
57346    157777     :___________:         :_____________:

            The approximate numbers on the left are called
     addresses   They are usually  shown  as  octal numbers
```

Figure 1. LSI-11 Memory Layout

available at each terminal. The total of these numbers does not include the area labeled *header* but includes the four remaining areas labeled *user code, symbol table, strings,* and *arrays.* The boundaries are not fixed, but shift as the program is altered or run.

The *header* for each user's job contains a list of the exact boundaries of the five areas at any point in time, the address of any free space, the name of the program, and the number of current lines being executed. It also contains work areas (buffers) which hold data input when editing a line in an immediate mode. Many other pointers, counters, and condition flags also can be found there.

The *user code* contains the intermediate form of code stored as tokens, text and references to the symbol table. All line numbers, variable names, numeric constants, and keywords are changed from the form in which they are entered to an altered form. Line numbers and variable names are stored as a two-byte reference to a spot in the symbol table. Numeric constants are stored in three different ways. If the number is an integer between 0 and 255, it is stored as two bytes (byte one is 375). If it is an integer between -32767 and + 32767 not in the range 0-255, it is stored as three bytes (byte one is 376), and if it is neither of the first two, it is stored as a first byte of 374, followed by a four-byte floating point number. Keywords are stored as a single byte token. Each statement ends with an end-of-line token regardless of whether this results from a backlash in the code or the termination of a line of code. The last byte of the user code area is an end-of-program token.

The *symbol table* contains one entry for each different line number and variable name in the program regardless of how many times either appears in the program. The line number reference consists of four bytes, two of them hold the line number and the other two point to the location of the beginning of the line in the user code area. The variable name may be numeric, or string, and may also represent an array. The variable entry requires ten bytes, the first two of which identify the type of variable it is, numeric (375 377), array (376 377) or string (377 377). The remaining bytes hold the variable name and either the value of a variable or point to where the value may be found. Every time the user code is scanned during execution, it must go to the symbol table to check out any reference to a line number or to check the current value of a variable.

```
                         Before Translation

10 IF X1<=9 THEN X1=515

                         After Translation

User Code Area (as octal bytes)

  10      IF    X1    <=    9   THEN   X1    =      515     \  END
000 000  205 000 004 257 375 011 255 000 004 267 376 002 003 200 232


Symbol Table (as octal bytes)

  LINE 10                   NUMERIC VARIABLE X1
000 012  xxx xxx   375 377  xxx xxx xxx xxx  000 000  130 061
  10     ADDRESS   NUMERIC       VALUE        UNUSED     X1

   All numbers shown in memory are shown in octal.   A line
   number  or variable reference in the user code area is a
   relative  address  in the symbol table.  Address 000 000
   is the first byte  of the table and 000 004 is the fifth
   byte.  The first two bytes  of  the  line  number in the
   symbol table is the only place the actual line number is
   permanently  recorded  anywhere  in  the total job area.
   The address portion (last two bytes) of  the line number
   entry is the absolute memory address  of  the first byte
   of the translated line in the user code area.
```

Figure 2. Sample view of user code and symbol table in memory.

The area containing *strings* holds the data pointed to by the symbol table whenever a string variable is referenced. Elements of string arrays are also contained in the string area. The symbol table data contains the length of a string variable. If much string manipulation is done in the program, the string area will become a scratch pad for string values that will eventually overflow. Then a 'garbage collector' routine will be called to consolidate all strings that are currently in use. This routine will also move the strings to one end of the string area so that free memory can be obtained in order to continue execution.

The values of the elements of numeric *arrays* are held in a separate area. If the array has more than ten elements, it is set up before execution of the user code as part of a preliminary scan of the code. This scan locates any DIM statement, so as to set up and initialize large arrays.

EDITING THE PROGRAM

Typing a line of code into the terminal in immediate mode results in a module known as the editor, calling a routine known as the translator. The translator constructs a new line of code from the line that was input.

ADVANCED COMPUTER PRODUCTS

THE FIRST TO OFFER PRIME PRODUCTS TO THE HOBBYIST AT FAIR PRICES NOW LOWERS PRICES EVEN FURTHER!

1. Proven Quality Factory tested products only, no re-tests or fallouts. Guaranteed money back. We stand behind our products.

2. Same Day Shipment All prepaid orders with cashiers check, money order or charge card will be shipped same day as received.

S100 PRODUCTS

LOGOS I 8K STATIC RAM
- ★ Low Power
- ★ Selectable Memory Protect
- ★ Totally Buffered
- ★ Phantom
- ★ Battery Back-up
- ★ Address on 1K boundary
- ★ Requires no front panel
- ★ No wait states

ASSEMBLED & TESTED
250ns. 199.95
450ns. 179.95

KIT
250ns. 149.95
450ns. 125.95

Bare PC Board w/Data **$21.95**
Now over 1 year successful field experience "Special Offer" Buy (4) 8K 450ns. Kits $117.00 ea.

IMS 16K STATIC RAM
- ★ Memory mapping
- ★ Phantom
- ★ Address 16K Bound
- ★ Use with or w/o front panel
- ★ Power 2.1 amps typ.
- ★ Uses 2114

ASSEMBLED & TESTED ONLY
450ns. $439.95 250ns. $495.95

S-100 32K STATIC RAM
- ★ Address 32K Boundary
- ★ Power 450ns. 2.8 amps typ.
- ★ No wait states on 2MHz
- ★ Fully Buffered
- ★ Phantom can be added
- ★ 2114/TMS4045 or 9135

ASSEMBLED & TESTED	KIT
250ns. $849.00	250ns. $790.00
450ns. $799.00	450ns. $740.00

Bare PC Board w/Data $59.95
SPECIAL OFFER:
Kit without Memory only $99.95. Add 2114 Memory for $160.00 per 8K

MICROBYTE 32K STATIC RAM
- ★ Phantom ★ TMS4044
- ★ Addressable 4K Blks ★ Independed reg.

ASSEMBLED & TESTED ONLY
450ns. $775.00
250ns. $875.00
(Special offer 16K version (expandable) 450ns. $435.00)

DC HAYES DATA COMMUNICATIONS ADAPTER
- ★ Telephone/TWX ★ S-100 compatible
- ★ Bell 103 freq. ★ Originate & answer mode

Assembled & Tested $279.95
Bare PC Board w/data .... $ 49.95

Z-80/Z-80A CPU BOARD
- ★ On board 2708
- ★ Power on jump
- ★ 2708 included (450ns.)
- ★ completely socketed

Assembled and tested $185.00
Kit $129.95
Bare PC Board .... $ 34.95
★ For 4MHz Speed Add $15.00

TDL COMPATIBLE Z-80 CPU
★ Compatible to TDL software & hardware ★ Can be upgraded for 4MHz operation

Assembled and tested $199.95
Kit $139.95
Bare PC Board .... $ 34.95

BYTE USER 8K EPROM BOARD
★ Power on Jump ★ Reset Jump

Assembled & Tested 94.95
Kit 64.95
Bare Pc Board 21.95
Special Offer: Buy 4 kits only $59.95 each

8080A CPU BOARD
S100 compatible with 8 level Vector Interrupt fully socketed.

Assembled & Tested $149.95
Kit .... $ 99.95
Bare PC Board w/data .... $ 34.95

TARBELL FLOPPY INTERFACE
- ★ S100 Compatible ★ Uses CPM
- ★ Jumper Selectable ★ Persci,Shugart,etc

Assembled and tested $269.95
Kit $179.95
Bare PC Board .... $ 39.95
NOTE: For CPM Add $70.00
Documentation Add $20.00

WMC REAL TIME CLOCK
- ★ S100 Compatible
- ★ Two interrupts
- ★ Software programmable
- ★ 1MHz Crystal Oscillator

Assembled and tested $179.95
Kit $124.95
Bare PC Board w/data .... $ 29.95

MICRODESIGN MR-16 2716 EPROM BOARD
- ★ Individual Prom Address
- ★ Uses Low cost 16K TI EPROMS
- ★ Optional 1K RAM ★ Phantom control

Assembled and Tested 174.95
Kit 99.50

DATABOOKS & MANUALS

NSC TTL Data	$ 3.95	AMI MOS/LSI Data	3.95	Mot. Vol 4 Mecl Data	3.95
NSC Linear Data	4.95	GI MOS/LSI Data	4.95	Mot. Vol. 5 CMOS Data	2.95
NSC Linear APP/note I	3.95	Osborne Intro to Micro Vol. 0	7.50	Mot. Vol. 6 Linear Data	3.95
NSC Linear APP/note II	3.95	Osborne Intro to Micro Vol. I	7.50	Mot. Vol. 9 Schottky TTL	2.95
NSC CMOS Data	2.95	Osborne Intro to Micro Vol. II	7.50	Mot. MPU Applications	25.00
NSC Audio Data	3.95	Osborne 8080 Programming	7.50	Mot. MPU Prog. Ref. Manual	3.95
NSC Volt. Reg. Data	2.95	Osborne 6800 Programming	7.50	Mot. Power Data	2.95
NSC Memory Data	3.95	Osborne Z80 Programming	7.50	Mot. Rectifier Data	2.95
NSC MOS/LSI Data	3.95	TI Power Semi's Data	7.50	Mot. Switching Tran	2.95
NSC Power Transistors	2.95	TI TTL Data	6.95	Mot. Zeners	2.95
Intel Databook	3.95	TI Transistors & Diodes	6.50	Basic Software SRI Vol. I or II	24.95
Intel MCS85 Manual	7.50	TI Memory Data	3.95	Basic Software SRI Vol. III	39.95
Intel MCS80 Manual	7.95	TI Optoelectronics	3.95	Basic Software SRI Vol. IV or V	9.95
Intel MCS40 Manual	4.95	TI Linear Data	3.95	Basic Software SRI Vol. VI	49.95
AMD 8080 Handbook	5.95	TI Bipolar Memory	3.95	Basic Software SRI Vol. VII	39.95
AMD Linear Data	4.95	TI Interface Data	4.95	1978 IC Master	47.50
AMD Schottky Data	4.95	Motorola Semi Data 1, 2, 3	9.95		

6800 DESIGNER BOARDS MODULES PROTO BOARDS

★ Motorola Compatible Modules*

MEK 6800 D2 Kit	$235.00
9600 6800 MPU Module	495.00
9601 16 slot Mother Bd.	175.00
9602 16 slot Card Cage	75.00
9603 8 slot Mother Bd.	99.00
9604 System Power Supply	250.00
9610 Prototype Board	36.00
9615 4KEprom Module	250.00
9620 16 port parallel I/O	375.00
9626 8K Static RAM	295.00
9626K 8K Static RAM Kit	225.00
9630 Extender Card	60.00
9640 Multiple Tuner Prog.	395.00
9650 8 port Duplex Asyn. I/O	395.00
Mot 43/86 Connectors w/w or s/t	5.95
AMI EVK 99* 6800 sub Kit	145.00
AMI EVK 100 Kit	269.95
AMI EVK 200 Kit	449.95
AMI EVK 300 Assembled	649.00
EVK Kluge Board	95.00
EVK 16K Byte Ram Board	75.00
EVK 6 Slot Motherboard	35.00
EVK Extender Board	45.00
EVK Solid Frame Chassis	129.00
EVK Connectors	6.50
AMI 6800 Proto Rom	30.00
AMI 6800 Micro Assembler Rom	30.00
6800 Tiny Basic Paper Tape	20.00
6800 Tiny Basic Eprom	125.00

ZILOG COMPATIBLE BOARDS

Z-80 CPU	395.00
ZDC Disc Controller	395.00
MEM 16/65K Memory	595.00

SUPPORT DEVICES

3881 (Z80 PIO)	11.25
3881-4 (4MHz)	14.95
3882 (Z-80 CTC)	11.25
3882-4 (4MHz)	14.95
8205/74S138 Decoder	2.95
8212 8 bit I/O	3.25
8214 Priority Int	7.95
8216 Bus Driver	3.25
8224 Clock Gen	3.95
8224-4 (4MHz)	12.95
8226 Bus Driver	3.95
8T26 Bus Driver	2.39
8228 Sys. Control	6.95
8238 Sys. Cont	7.95
8251 Prog. I/O	11.50
8253 Int. Timer	27.95
8255 Prog. I/O	11.50
8257 Prog. DMA	27.95
8259 Prog. Int	27.95
8279 Prog. Keyboard	19.95
6810-1 128 x 8 RAM	4.75
6820 PIA	7.95
6828 Priority Int	11.95
6834-1 512 x 8 Eprom	12.95
6850 ACIA	7.95
6852 Serial Adapter	9.95
6860 Modem	14.50
6862 Modulator	17.50
6871A 1.0MHz OSC	25.95
2350 USRT	9.95
6880 Bus Driver	2.39
1821 SCD 1K RAM	25.00
1822 SCD 256 x 4 RAM	16.95
1824 CD 32 x 8 RAM	9.95
1852 CD 8 bit I/O	10.95
1856 CD I/O	8.95
1857 CD I/O	8.95
6520 PIA	8.95
6522 Mult	9.95
6530-002	15.50
6530-003	15.50
6530-004	15.50
6530-005	15.50
3851 F8 Prog. Store	14.95
3853 F8 Memory I/O	14.95

PROMS

2708	12.95
2708-6	5.95
1702A	4.95
1702A-6	3.75
2716-5V	44.95
2716-5V, 12V	24.95
5203AQ	6.95
5204AQ	9.95
6834-1	12.95
IM 5610	2.95
82S123	2.95
82S126	4.50
82S129	2.95

RAMS DYN.

4116 16K x 1	27.95
4115 8K x 1	14.95
4050 4K x 1	4.45
4060 4K x 1	4.45
4096 4K x 1	4.25
2104 4K x 1	4.95
2107B-4 4K x 1	4.25
5261	
5262	1.95
5270	4.95
5280	4.95
1103	1.95
4008L	2.95
TMS4070 16K x 1	19.95

FLOPPY DISK I/O

1771 I/O 8" only	22.95
1771-01 8" & Minifloppy	35.95
uPd372 Nec Floppy	49.95
1781 Dual Floppy	89.95

CHARACTER GENERATORS

RO32513-001 (5V) Upper	9.50
RO32513-002 (5V) Lower	10.95
RO32513-ADM3 (5V) Lower	14.95
MCM6571	10.75
MCM6571A	10.75
MCM6574	14.50
MCM6575	14.50

KEYBOARD ENCODERS

AY5-2376	13.75
AY5-3600	13.75

UARTS/USRTS

TR1602B (5V, 12V)	3.95
AY51013 (5V, 12V)	5.50
AY51014A/1612 (5-14V)	7.95
AY51015A/1863 (5V)	7.95
TMS 6011 (5V, 12V)	5.50
IM6402	9.95
IM6403	9.95
2350 USRT	9.95
1671B Astros	24.95

BAUD RATE GENERATORS

MC14411	11.25
4702	14.95
WD1941	9.95

TV CHIPS

MM5320 TV Synch	7.95
MM5369 Prescaler	3.95
LM1889 Modulator	3.95
CW300 Saw Function	1.50
MM57100 Color TV Game	6.95
MM57104 Clock	3.75
AY38500-1 TV Game	5.95
AY38600 Color TV Game	24.95
AY38615-1 Color Converter	8.95
AY38700-1 Tank Chip	29.95
RF Modulator with Audio	8.95

A/D CONVERTERS

8700 8 bit Binary	13.50
8701 10 bit Binary	22.00
8702 12 bit Binary	36.95
8705 12 bit TS	42.95
9400 Volt to Freq. Conv.	7.25
8750 3-1/2 Digit BCD	13.95
8038 Multifunction	3.95
1408L6 6 bit	5.95
1408L8 8 bit	3.95

MPU & COMPUTER KITS

MEK 6800 D2 Kit	235.00
KIM I 6502	245.00
Intercept Jr 6100	281.00
Technico 9900 Kit	299.00
RCA VIP Cosmac	275.00
AMI EVK 99	145.00

MICROPROCESSORS

Z-80	**$21.95**
Z-80A	29.95
F-8 (3850)	19.95
2650	24.95
CD1802	19.95
8080A	12.95
8085	29.95
8008-1	14.95
2901	21.95
2901A	29.95
TMS 9900JL	89.95
CP1600	39.95
6502	19.95
IM6100	29.95
6800	**19.95**
6802P	32.95

SOCKETS

8 Pin ww	32	14 Pin S/T	20
14 Pin ww	37	16 Pin S/T	22
16 Pin ww	38	18 Pin S/T	31
18 Pin ww	60	22 Pin S/T	35
20 Pin ww	90	24 Pin S/T	41
22 Pin ww	93	28 Pin S/T	49
8 Pin S/T	17	40 Pin S/T	63

CONNECTORS

6 Pin Single S/E	1.49
15/30 Dual S/E	1.95
18/36 Dual S/E	2.35
22/44 Dual S/E	2.95
43/36 Dual S/T	6.50
43/86 Dual W/W	6.50
50/100 IMSAI W/W	4.75
50/100 IMSAI S/T	4.75
50/100 Altair W/W	5.95
50/100 Altair S/T	5.95

25 Pin D Subminature
DB25P 3.25
DB25S 3.75
Hood 1.25

IMSAI Card Guides 4/1.00

RS232 CONNECTOR SET W/Hood $6.50

NEW CTS DIPSWITCHES

CTS206-4	$1.75	CTS208-8	$1.95
CTS206-5	$1.75	CTS209-9	$1.95
CTS206-6	$1.75	CTS209-10	$1.95
CTS207-7	$1.75		

We also stock full of 7400, 74LS, 74L Linear and CMOS Send for pricing or use our competitors STD catalog pricing.

NAKED PC BOARD SALE S-100

Z-80 CPU (ITHACA)	34.95
Z-80 CPU (TDL comp.)	34.95
8080A CPU	34.95
8K Static Logos	21.95
8K Eprom Byteuser	21.95
32 Static S-100	59.95
Tarbell Floppy I/O	39.95
Tarbell Cassette	29.95
Vector 8800 Proto	19.95
ACP Proto Bd	27.95
ACP Extender w/conn	15.95
Realtime Clock	34.95
WMC 13 Slot Mother Bd	33.00
S-100 8 Slot Mother Bd	34.95
Vector 8803 11 Slot M Bd	29.95
SSM Video I/O	25.95
SSM I02 Parallel I/O	25.95

UV "Eprom" Eraser
Model UVs-11E **$59.50**
Holds 4 Eprom's at a time.
Backed by 45 years UV experience
Model S-52T **$219.95**
Professional Industrial Model

KIMSAI
Expand your KIM-1 to S-100
Kit $125.00 Assembled $165.00

SPECIAL DIGITAL CASSETTES 3/$4.25
High quality cassettes manufactured for digital application

TV INTERFACES

Pixie-Verter	8.50
TV-1 Video Interface	8.95
Pickles & Trout TV Mod. Kit	20.00

NOTICE
We are looking for software support packages to offer in our new catalogue. Please write or call if you want to participate.

All Shipments FCM or UPS. Orders under $100.00 add 5% handling and postage. Orders over $100.00 add 2.5% handling & postage. Masterchange/Bankamericard/COD accepted w/25% deposit. California Residents add 6% tax. Foreign Orders add 8% handling. All parts prime factory tested guaranteed. Same day shipment. Add 25 cents for Data.

Retail pricing may vary from Mail Order Pricing.

STATIC RAM HEADQUARTERS

	1-24	25-99	100
21L02 450ns	1.30	1.25	1.15
21L02 250ns	1.59	1.55	1.45
2102	1.25	1.15	1.10
91L02APC	1.75	1.65	1.50
21L11-1	4.10	3.95	3.75
2111-1	3.75	3.65	3.55
21L12-1	4.25	4.10	3.95
2112-1	2.95	2.85	2.65
21L01-1	3.95	3.75	3.65
2101-1	2.90	2.70	2.55
2114-3 300ns	12.95	10.95	9.95
2114-4 450ns	11.25	10.25	9.50
TMS4044	11.25	10.25	9.50
TMS4045	11.25	10.25	9.50
EMM4200A	12.50	11.95	9.95
EMM4402	9.95	9.95	8.25
5101C-E	10.95	10.50	9.90
upd410 (4200)	10.95	10.25	9.25
AMD9140/41	10.95	10.25	9.25
AMD9130/31	12.95	11.95	10.25
FSC 460/46416K CCD Only $18.95 Each			
1101	1.95	1.75	1.25

WAVEFORM GENERATORS

8038 Function Gen	3.95
MC4024 VCO	2.45
LM566 VCO	1.75
XR2206 Function Generator	5.25

CHARGE COUPLED DEVICES
16K CCD First time offered Fairchild 460 CCD 16K Memory (now you can experiment with CCD technology at a reasonable price 17 page Application note supplied with each order. Quantity Limited)
$18.95 each (reg. 43.00)

CRYSTALS
Microprocessor Timebases TV Game

Frequency	Application	Price
1.0MHz	6800/S6860	$5.85
1.8432	MC14411	4.95
2.0MHz	G-8/TV Games	5.85
2.01MHz	TV Games	2.95
2.097152MHz	MM5378/79 Auto	5.85
2.4576MHz	34702 Bit Rate Gen	5.85
3.579545MHz	MM5369 (TV Color)	1.50
4.0MHz	4004/4040	4.95
4.194304MHz	8038	5.95
4.91520MHz	Std. CLK Freq	5.95
5.0MHz	Several MPU's	4.95
5.0688	COM5016	4.95
5.7143MHz	PACE	5.95
6.0MHz	Misc	4.95
6.144	Intel 80885	4.95
6.5536	ICM7045	4.95
10.0MHz	Several MPU's	4.95
13.0MHz	Misc	4.95
14.31818	Misc	4.95
18.0MHz	8080/8008/8224	4.95
18.432MHz	8080/8008/8224	5.95
20.0MHz	Several MPU's	4.95
22.1184MHz	8080A	5.95
27.0MHz	8080A	5.95
36.0MHz	8080A	5.95
48.0MHz	TIM9904	5.95
100KC	Misc	12.95

DISPLAYS/OPTO

DL 704/707/CC/CA 300	1.25
FND359 CC 357	.95
FND 500/507/CC/CA 500	1.35
FND 503/510 CC/CA 500	.95
FND 800/807 CC/CA 800	2.50
Bowmar 9 digit bubble	.99
FSC 8024 4 digit CC 800	4.95
HP7340 HEX Display	15.95
TIL 305 5 x 7 Array	4.50
TIL 306 7 seg w/logic	8.95
TIL 308 7 seg w/logic	8.95
TIL 309 7 seg w/logic	7.95
TIL 311 HEX Display	9.25
MA 1003 12 auto clock	17.95
MA 1002 4 digit clock module	9.95
MA 1010 4 digit clock module	9.95
NSN 373/374 dual CC/CA 300	2.20
NSN 583/584 ddual CC/CA 500	2.60
NSN 783/784 dual CC/CA 700	3.00
4N25 Opto Isolater	3/1.99
MCT 2 Opto Isolater	.89
4N33 Darlington ISOL	1.75
Red Led's 185 Dia	5/1.00
Green/Yellow	4/1.00

MONTHLY SPECIALS

14 Pin S/T sockets 1000 pcs	.10
16 Pin S/T sockets 1000 pcs	.11
22 Pin S/T sockets 1000 pcs	.12
FSC460/464 16K CCD's	18.95
AMD 9140CDC Static Ram	10.95
4116/416 16K Dynamic Ram's	8 for/200.00
1488 or 1489 RS232 Driver	1.25
8T97 Buffer	1.25
8130 or 8131	2.45
8833 or 8835	1.99
74367 or 74368	6/5.00
75451/52/53	10/2.50
8T26 Bus Driver	2.39
78L05 Regulator	2/1.00
4N25 Optoisolator	3/1.99
MCT2 Optoisolator	.89
2510A Shift Register	2.95
2519A Shift Register	2.95
MM5056 Shift Register	2.95
LM760N-14 Comparitor	1.10
DS0025CN Clock Driver	1.49
DS0026CN Clock Driver	2.50
8T20 Bidirect oneshot	3.49

NEW 1978 CATALOGUE AVAILABLE FEB. SEND .25 POSTAGE

DISCOUNT COMPUTER CORNER

	Cash	Credit Card
Sorcc IQ 120	895.00	925.00
Imsai w/22 slots	629.00	647.00
Speechlab	269.00	278.00
Cromemco Z-2	536.00	551.00
Horizon 1	1439.00	1479.00
North Star	599.00	617.00
TEI MCS122	450.00	462.00
Apple w/16K	1198.00	1234.00
Decwriter II	1475.00	1525.00
Computalker	365.00	380.00
Persci 277	1230.00	1266.00
Shugart 800-2	575.00	590.00
Low cost S100 Bus	149.95	155.95

P. O. BOX 17329 Irvine, California 92713 New Phone (714) 558-8813
TWX: 910-595-1565

Retail Store Open Mon. – Sat. Located at 1310 "B" E. Edinger, Santa Ana, CA 92705

This new line will be inserted into the user code area in memory by the editor.

The translator analyzes the line character by character, deciding what token to use and checking the symbol table for line numbers and variable names. Figure 3 shows a simplified flow chart that illustrates the translation process. If the translator does not find that an entry already exists in the table, it creates an entry at the end of the table. It alters the form in which numbers are stored, as mentioned earlier. Special text tokens are also inserted between quotes and text or in a remark statement. If a NEXT statement is found, the translator sets up a ten-byte work area to aid in execution of the code at a later time.

After the translation is complete, control returns to the editor routine. This routine inserts new and modified lines or deletes lines whose line number is typed alone, by updating the translated code in the user code area. The entire symbol table must be moved each time there is a change in size of user code. Because the 'free area' of memory is located between the symbol table and the string area, the boundaries of both areas may float when room is needed during the typing in of code or during the execution of the program.

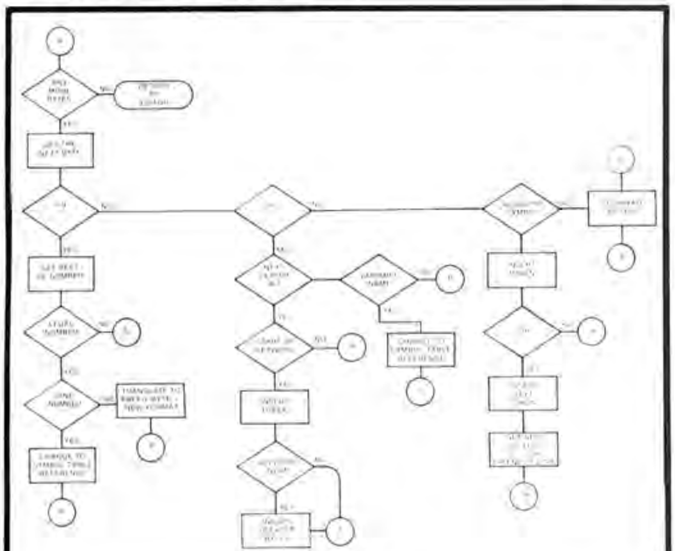


Figure 3.

RUNNING THE PROGRAM

When the command is given to RUN the program, MUBAS will become very busy. We will ignore all the times it might be interrupted by the operating system to process the ticking of a clock, the processing of I/O, and the swapping of control from one user to the next. Instead, we will concentrate on the execution of the translated user code which is waiting in memory. First, the editor has the command translated just as it would for a numbered statement. When it realizes there is no line number, it will look the RUN command up in a table of commands to be sure it is valid. It then passes control to the execution module. This modules makes sure you didn't change your mind and hit control C key (^C) to stop the program, and then dispatches the command to be executed by a RUN routine.

This run routine does two main types of tasks. It cleans up any unfinished business from the last time the program was run (we may have just been in the process of debugging), and sets up new parameters based on the current state of the program. It must tidy up any open files and eliminate the buffer space allotted to such files. It must also clear out the values of any variables both in the symbol table and the string area, and reset any parameters associated with the random number generator or special keyboard characteristics.

After these items have been initialized, the RUN routine then starts to examine the code in the user area to locate any DIM statement. Any DIM statement found is used to allocate room in the array area, if the array is numeric. DEF statements are also examined so as to establish any user defined functions needed in the program. The presence of a RANDOMIZE statement will uniquely initialize the parameters of the RND function, and then control will pass back to the execution routine

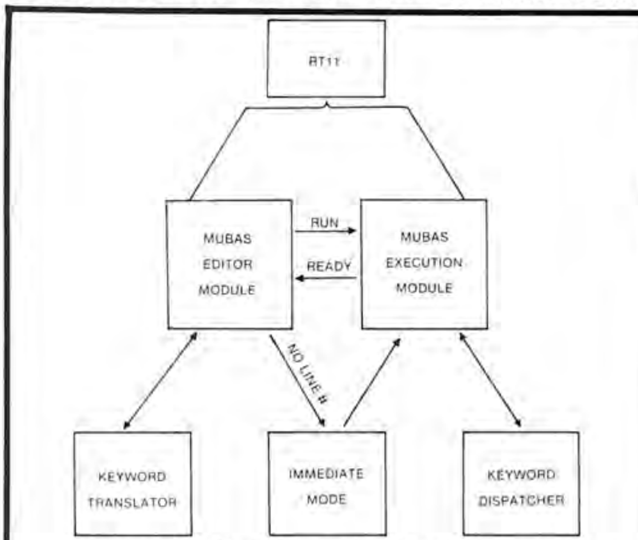


Figure 4. Overview of Translating and Dispatching Keywords.

with the line number pointer set to begin execution at the first statement of the program.

The execution routine will then check for ^C, and if execution is to continue, will start to unravel the tokens in the first statement. Since a large part of the syntax checking has already been done when the token was originally generated, the execution can take place very quickly. However, even the simplest of statements, such as a GOTO statement, would require dozens of instructions to be executed. MUBAS scans the translated line until it finds the GOTO token. This must be checked in a table of tokens and control is passed to the routine that handles GOTO's. This routine calls another routine to get the next two bytes and use them to get the line number address from the symbol table. Some error checking must be done in case the line number address is not in the symbol table, or an end of line token does not follow the code already interpreted. The pointer that indicates the current line being executed must be updated, and control then passes back to the execution module.

We can see from this discussion that an interpretive compiler like MUBAS can not compete, in terms of sheer speed, with a truly compiled language in which a GOTO statement requiring dozens of statements could be replaced by a single jump statement (JMP). However, the convenience factor of the flexibility of the interactive debugging of a program, and the simplicity of use, make BASIC a very powerful language. A second important consideration is that for most applications outside of scientific computing, the issue of speed does not center around the speed of execution in memory, but rather on the slowness of peripherals like disks or printers. In other words, most applications are I/O bound and not compute bound. This can truly turn the tokens in BASIC into tokens of good fortune! □

LOOK

By Kenyon Swartwout

From time to time, it is necessary or desirable to make modifications to an existing assembly language program, such as BASIC, in order to accomplish something that it was not originally designed to do. However, even after disassembly of the program, it takes a time-consuming search to locate the section of the program where the changes need to be made.

The LOOK program will simplify and speed up this locating task. Merely determing the byte or sequence of bytes that you are looking for and the LOOK program will locate each place that this byte or sequence of bytes appears in the unknown program and will then output each of these addresses together with the sequence of byte. If the sequence of bytes is not found in the unknown program, NO MATCH will be displayed. The program is self-prompting and will ask the following questions:

HOW MANY BYTES (1-9)?
LIST BYTES IN HEX:
START ADDRESS:
FINISH ADDRESS:

The program has been designed to work with an 8080 microcomputer. The equipment I am using consists of an IMSAI computer, a North Star floppy disk, a National Multiplex I/O board and a SOROC video terminal. The program is addressed at 0000, but could be assembled to operate at any memory location. For other locations, lines 3250 and 3260 must be changed to relocate the Origin and Stack Pointer.

For use with other types of equipment, the input routine at line 2010 and the output routine at line 2210 may have to be changed. If the program is always used with the North Star DOS, the input and output routines can be eliminated by calling the DOS input and output routines.

While inputting data, a CONTROL C will stop the program and return it to the start. When asking for the number of bytes, a "0" will return the computer to the North Star DOS. This happens at line 0170. After completing each search, the program will return to the start and again ask for the number of bytes.

As an example of how the program operates, assume that you want to find every location where the input routine (CD 95 01) is called within our LOOK program. Here is how it would go:

HOW MANY BYTES (1-9)? 3
LIST BYTES IN HEX: CD 95 01
START ADDRESS: 00 00
FINISH ADDRESS: 02 32

0011 CD 95 01
0042 CD 95 01
004B CD 95 01

HOW MANY BYTES (1-9)?

This tells us that the byte sequence of CD 95 01 will be found at addresses 0011, 0042 and 004B.

If any reader would like to have this program recorded on a North Star disk, please send me a blank disk, together with $10.00, and the disk will be loaded with the North Star DOS, the LOOK program assembled at 0000 and the unassembled program, LOOK1. The disk will be returned within 48 hours. The unassembled LOOK1 program can be modified as outlined above and then used to assemble the program at any desired location. □

PROGRAM LISTING

```
A 0000

0000              0010 *THIS PROGRAM WILL LOOK FOR A BYTE OR
0000              0020 *A SEQUENCE OF 2-9 BYTES IN ANY PROGRAM
0000              0030 *AND WHEN FOUND, THE ADDRESS AND BYTE
0000              0040 *WILL BE PRINTED.  IF NO MATCH IS FOUND
0000              0050 *'NO MATCH' WILL BE PRINTED. TYPE A '0'
0000              0060 *WHEN ASKED FOR # OF BYTES TO RETURN TO DOS.
0000              0070 *CONTROL 'C' RETURNS TO START
0000              0080 *
0000 31 00 04     0090 START  LXI   SP,STACK
0003 21 DF 01     0100        LXI   H,MA
0006 CD C3 01     0110        CALL  CRLF
0009 CD 26 02     0120        CALL  ZBUF       ZERO BUFFER
000C CD 1A 02     0130        CALL  STOUT      OUTPUT 1ST MESSAGE
000F 23           0140        INX   H          TO NEXT MESSAGE
0010 E5           0150        PUSH  H          SAVE ADDRESS
0011 CD 95 01     0160 ST1    CALL  CHIN
0014 FE 30        0170        CPI   '0'
0016 CA 28 20     0180        JZ    DOS
0019 FE 31        0190        CPI   '1'
001B DA 11 00     0200        JC    ST1
001E FE 3A        0210        CPI   '9'+1
0020 D2 11 00     0220        JNC   ST1
0023 47           0230        MOV   B,A
0024 CD BB 01     0240        CALL  CHOUT      OUT NUMBER
0027 D6 30        0250        SUI   48
0029 4F           0260        MOV   C,A        SAVE COUNT IN C
002A 32 00 04     0270        STA   NBR
002D 32 0E 04     0280        STA   NM
0030 CD C3 01     0290        CALL  CRLF
0033 E1           0300        POP   H
0034 CD 1A 02     0310        CALL  STOUT      OUTPUT 2ND MESSAGE
0037 23           0320        INX   H          TO 3RD MESSAGE
0038 E5           0330        PUSH  H          SAVE ADDRESS
0039 11 02 04     0340        LXI   D,BYTE
003C CD 42 00     0350        CALL  LK1
003F C3 69 00     0360        JMP   LK2
0042 CD 95 01     0370 LK1    CALL  CHIN
0045 47           0380        MOV   B,A
0046 CD BB 01     0390        CALL  CHOUT
0049 12           0400        STAX  D          PUT 1ST CHAR IN MEMORY
004A 13           0410        INX   D
004B CD 95 01     0420        CALL  CHIN       INPUT 2ND CHAR.
004E 47           0430        MOV   B,A
004F CD BB 01     0440        CALL  CHOUT      OUTPUT 2ND CHAR.
0052 12           0450        STAX  D          PUT 2ND CHAR IN MEMORY
0053 06 20        0460        MVI   B,' '
0055 CD BB 01     0470        CALL  CHOUT
0058 1B           0480        DCX   D          BACK TO 1ST CHAR.
0059 CD FB 01     0490        CALL  AHEX       HEX TO BINARY
005C 7D           0500        MOV   A,L
005D 1B           0510        DCX   D
005E 1B           0520        DCX   D
005F 12           0530        STAX  D
0060 13           0540        INX   D
0061 0D           0550        DCR   C
0062 3E 00        0560        MVI   A,0
0064 B9           0570        CMP   C
0065 C8           0580        RZ    RETURN     WHEN ENTERED
0066 C3 42 00     0590        JMP   LK1        TO ENTER NEXT BYTE
0069 E1           0600 LK2    POP   H
006A CD C3 01     0610        CALL  CRLF
006D CD 1A 02     0620        CALL  STOUT      OUT START ADDR MESSAGE
0070 23           0630        INX   H
0071 E5           0640        PUSH  H
0072 11 0F 04     0650        LXI   D,SA
0075 0E 02        0660        MVI   C,2
0077 CD 42 00     0670        CALL  LK1        TO ENTER START ADDR.
```

```
007A CD BB 01         0680         CALL  REV
007D E1               0690         POP   H
007E CD C3 01         0700         CALL  CRLF
0081 CD 1A 02         0710         CALL  STOUT        OUT FIN. ADDR MESSAGE
0084 11 12 04         0720         LXI   D,FA
0087 0E 02            0730         MVI   C,2
0089 CD 42 00         0740         CALL  LK1          TO ENTER FINISH ADDR
008C CD BB 01         0750         CALL  REV
008F                  0760 *
008F                  0770 *LOOK FOR CHARACTER SEQUENCE
008F                  0780 *
008F CD C3 01         0790         CALL  CRLF
0092 21 12 04         0800 LOOK    LXI   H,FA         FINISH ADDR IN H,L
0095 4E               0810         MOV   C,M
0096 23               0820         INX   H
0097 46               0830         MOV   B,M
0098 11 02 04         0840 LOOK1   LXI   D,BYTE
009B 2A 0F 04         0850         LHLD  SA
009E 1A               0860         LDAX  D
009F F5               0870         PUSH  PSW
00A0 3A 00 04         0880         LDA   NBR
00A3 32 01 04         0890         STA   NUM
00A6 CD 60 01         0900         CALL  ENCK         CHECK FOR FINISH
00A9 F1               0910         POP   PSW
00AA BE               0920         CMP   M
00AB CA B5 00         0930         JZ    LOOK2
00AE 23               0940         INX   H
00AF 22 0F 04         0950         SHLD  SA           NEXT ADDR IN SA
00B2 C3 98 00         0960         JMP   LOOK1
00B5 23               0970 LOOK2   INX   H
00B6 22 0F 04         0980         SHLD  SA
00B9 2B               0990         DCX   H
00BA CD 7D 01         1000 LOOK3   CALL  FIN
00BD 23               1010         INX   H
00BE 13               1020         INX   D
00BF 1A               1030         LDAX  D
00C0 BE               1040         CMP   M
00C1 CA BA 00         1050         JZ    LOOK3
00C4 C3 98 00         1060         JMP   LOOK1
00C7                  1070 *
00C7                  1080 *THIS ROUTINE PRINTS ADDRESS AND MATCHED BYTES
00C7                  1090 *
00C7 33               1100 PRT     INX   SP
00C8 CD C3 01         1110         CALL  CRLF
00CB 2A 0F 04         1120         LHLD  SA
00CE 2B               1130         DCX   H            ADDR OF OK BYTE
00CF 22 0F 04         1140         SHLD  SA
00D2 7C               1150         MOV   A,H          CONVERT TO HEX & OUT
00D3 E5               1160         PUSH  H
00D4 CD CE 01         1170         CALL  HOUT
00D7 E1               1180         POP   H
00D8 7D               1190         MOV   A,L
00D9 CD CE 01         1200         CALL  HOUT
00DC 06 20            1210         MVI   B,' '
00DE CD BB 01         1220         CALL  CHOUT
00E1 21 00 04         1230         LXI   H,NBR
00E4 4E               1240         MOV   C,M
00E5 21 02 04         1250         LXI   H,BYTE
00E8 7E               1260 PRT1    MOV   A,M
00E9 E5               1270         PUSH  H
00EA CD CE 01         1280         CALL  HOUT
00ED E1               1290         POP   H
00EE 06 20            1300         MVI   B,' '
00F0 CD BB 01         1310         CALL  CHOUT
00F3 0D               1320         DCR   C
00F4 E5               1330         PUSH  H
00F5 21 0E 04         1340         LXI   H,NM
00F8 79               1350         MOV   A,C
00F9 71               1360         MOV   M,C
00FA E1               1370         POP   H
00FB 3E 00            1380         MVI   A,0
00FD B9               1390         CMP   C
00FE CA 05 01         1400         JZ    PRT2         IF THRU
0101 23               1410         INX   H
0102 C3 E8 00         1420         JMP   PRT1
0105 2A 0F 04         1430 PRT2    LHLD  SA
0108 23               1440         INX   H
0109 22 0F 04         1450         SHLD  SA
010C C3 92 00         1460         JMP   LOOK
010F 48 4F 57 20 4D   1470 MA      ASC   'HOW MANY BYTES (1-9)?'
     41 4E 59 20 42
     59 54 45 53 20
     28 31 2D 39 29
     3F
0124 0D               1480         DB    CR
0125 4C 49 53 54 20   1490         ASC   'LIST BYTES IN HEX:'
     42 59 54 45 53
     20 49 4E 20 48
     45 58 3A
0137 0D               1500         DB    CR
0138 53 54 41 52 54   1510         ASC   'START ADDRESS:'
     20 41 44 44 52
     45 53 53 3A
0146 0D               1520         DB    CR
0147 46 49 4E 49 53   1530         ASC   'FINISH ADDRESS:'
     48 20 41 44 44
     52 45 53 53 3A
0156 0D               1540         DB    CR
0157 4E 4F 20 4D 41   1550 MA1     ASC   'NO MATCH'
     54 43 48
015F 0D               1560         DB    CR
0160                  1570 *
0160                  1580 *THIS ROUTINE CHECKS FOR HL AT FINISH ADDRESS
0160                  1590 *
0160 78               1600 ENCK    MOV   A,B
0161 BC               1610         CMP   H
0162 C0               1620         RNZ                RETURN IF NOT COMPLETE
0163 79               1630         MOV   A,C
0164 BD               1640         CMP   L
0165 C0               1650         RNZ                RETURN IF NOT COMPLETE
0166 CD C3 01         1660         CALL  CRLF
0169 3A 0E 04         1670         LDA   NM
016C FE 00            1680         CPI   0
016E CA 00 00         1690         JZ    START
0171 21 57 01         1700         LXI   H,MA1
0174 CD 1A 02         1710         CALL  STOUT
0177 CD C3 01         1720         CALL  CRLF
017A C3 00 00         1730         JMP   START
017D                  1740 *
017D                  1750 *CHECK FOR COMPLETION OF BYTE MATCH
017D                  1760 *
017D 3A 01 04         1770 FIN     LDA   NUM
0180 3D               1780         DCR   A
0181 32 01 04         1790         STA   NUM          NEW # IN MEMORY
0184 CA C7 00         1800         JZ    PRT          JUMP TO PRINT IF THRU
0187 C9               1810         RET
0188                  1820 *
0188                  1830 *THIS ROUTINE REVERSES 2 BYTES IN MEMORY
0188                  1840 *
0188 1B               1850 REV     DCX   D
0189 EB               1860         XCHG
018A C5               1870         PUSH  B
018B 4E               1880         MOV   C,M
018C 2B               1890         DCX   H
018D 46               1900         MOV   B,M
018E 71               1910         MOV   M,C
018F 23               1920         INX   H
0190 70               1930         MOV   M,B
0191 C1               1940         POP   B
0192 EB               1950         XCHG
0193 13               1960         INX   D
0194 C9               1970         RET
0195                  1980 *
0195                  1990 *INPUT ROUTINE. ASCII CHAR RETURNED IN ACC.
0195                  2000 *
0195 DB 00            2010 CHIN    IN    0
0197 E6 01            2020         ANI   1
0199 C2 95 01         2030         JNZ   CHIN
019C DB 01            2040         IN    1
019E E6 7F            2050         ANI   7FH
01A0 FE 03            2060         CPI   3
01A2 CA 00 00         2070         JZ    START
01A5 FE 30            2080         CPI   '0'
01A7 DA 95 01         2090         JC    CHIN
01AA FE 3A            2100         CPI   '9'+1
01AC D8               2110         RC
01AD FE 41            2120         CPI   'A'
01AF DA 95 01         2130         JC    CHIN
01B2 FE 47            2140         CPI   'F'+1
01B4 D2 95 01         2150         JNC   CHIN
01B7 C9               2160         RET
01B8                  2170 *
01B8                  2180 *OUTPUT ROUTINE. ASCII CHAR MUST BE IN B REGISTER
01B8                  2190 *CHAR WILL ALSO BE IN ACC. ON RETURN
01B8                  2200 *
01B8 DB 00            2210 CHOUT   IN    0
01BA E6 80            2220         ANI   80H
01BC C2 BB 01         2230         JNZ   CHOUT
01BF 78               2240         MOV   A,B
01C0 D3 01            2250         OUT   1
01C2 C9               2260         RET
01C3                  2270 *
01C3                  2280 *CR-LF ROUTINE
01C3                  2290 *
01C3 06 0D            2300 CRLF    MVI   B,CR
01C5 CD BB 01         2310         CALL  CHOUT
01C8 06 0A            2320 LF      MVI   B,10
01CA CD BB 01         2330         CALL  CHOUT
01CD C9               2340         RET
01CE                  2350 *
01CE                  2360 *THIS ROUTINE CONVERTS A BINARY VALUE TO ASCII
01CE                  2370 *HEX AND OUTPUTS CHAR.  MEMORY AT HCON.
01CE                  2380 *
01CE CD DE 01         2390 HOUT    CALL  BINH
01D1 21 0C 04         2400         LXI   H,HCON       CONVERSION AREA
01D4 46               2410 CHOT    MOV   B,M
01D5 CD BB 01         2420         CALL  CHOUT
01D8 23               2430         INX   H
01D9 46               2440         MOV   B,M
01DA CD BB 01         2450         CALL  CHOUT
01DD C9               2460         RET
01DE                  2470 *
01DE                  2480 *CONVERTS BINARY VALUE IN REG A TO ASCII HEX
01DE                  2490 *DIGITS AND STORES IN MEMORY.
01DE                  2500 *
01DE 21 0C 04         2510 BINH    LXI   H,HCON
01E1 47               2520         MOV   B,A          SAVE VALUE
01E2 1F               2530         RAR
01E3 1F               2540         RAR
01E4 1F               2550         RAR
01E5 1F               2560         RAR
01E6 CD F1 01         2570         CALL  BIN1
01E9 77               2580         MOV   M,A
01EA 23               2590         INX   H
01EB 78               2600         MOV   A,B
01EC CD F1 01         2610         CALL  BIN1
01EF 77               2620         MOV   M,A
01F0 C9               2630         RET
01F1                  2640 *
01F1                  2650 *THIS ROUTINE CONVERTS BINARY TO HEX
01F1                  2660 *
01F1 E6 0F            2670 BIN1    ANI   0FH          LOW 4 BITS
01F3 C6 30            2680         ADI   48           CONVERTS TO ASCII
01F5 FE 3A            2690         CPI   58           DIGIT 0-9
01F7 D8               2700         RC
01F8 C6 07            2710         ADI   7            MODIFY FOR A-F
01FA C9               2720         RET
01FB                  2730 *
01FB                  2740 *THIS ROUTINE FETCHES DIGITS FROM THE
01FB                  2750 *BUFFER ADDRESSED BY B,C AND CONVERTS
01FB                  2760 *ASCII HEX DIGITS TO BINARY.  UP TO
01FB                  2770 *16 BITS CAN BE CONVERTED.  STOPS AT 0.
01FB                  2780 *
01FB 21 00 00         2790 AHEX    LXI   H,0
01FE 1A               2800 AHE1    LDAX  D            FETCH ASCII DIGIT
01FF B7               2810         ORA   A            SET Z FLAG
0200 C8               2820         RZ
0201 29               2830         DAD   H
0202 29               2840         DAD   H
0203 29               2850         DAD   H
0204 29               2860         DAD   H
0205 CD 12 02         2870         CALL  ASH1         CONVERT TO BINARY
0208 FE 10            2880         CPI   10H          CHECK FOR LEGAL VALUE
020A 3F               2890         CMC
020B D8               2900         RC                 RETURN IF ERROR
020C B5               2910         ADD   L
020D 6F               2920         MOV   L,A
020E 13               2930         INX   D
020F C3 FE 01         2940         JMP   AHE1
0212                  2950 *
0212                  2960 *CONVERTS ASCII HEX TO BINARY
0212                  2970 *
0212 D6 30            2980 ASH1    SUI   48           ASCII BIAS
0214 FE 0A            2990         CPI   10           DIGIT 0-10
0216 D8               3000         RC
0217 D6 07            3010         SUI   7            ALPHA BIAS
0219 C9               3020         RET
021A                  3030 *
021A                  3040 *THIS ROUTINE OUTPUTS CHARACTERS OF A
021A                  3050 *STRING UNTIL A CARRIAGE RETURN IS FOUND
021A                  3060 *
021A 46               3070 STOUT   MOV   B,M          FETCH CHAR
021B 3E 0D            3080         MVI   A,CR
021D B8               3090         CMP   B            CHAR=CR?
021E C8               3100         RZ
021F CD BB 01         3110         CALL  CHOUT
0222 23               3120         INX   H
0223 C3 1A 02         3130         JMP   STOUT
0226                  3140 *
0226                  3150 *THIS ROUTINE ZEROES OUT A BUFFER
0226                  3160 *
0226 AF               3170 ZBUF    XRA   A            GET A 0
0227 11 00 04         3180         LXI   D,NBR
022A 06 16            3190         MVI   B,22
022C 12               3200 ZBU1    STAX  D
022D 13               3210         INX   D
022E 05               3220         DCR   B
022F C2 2C 02         3230         JNZ   ZBU1
0232 C9               3240         RET
0233                  3250         ORG   0400H
0400                  3260 STACK   EQU   0400H
0400                  3270 NBR     DS    1
0401                  3280 NUM     DS    1
0402                  3290 BYTE    DS    10
040C                  3300 HCON    DS    2
040E                  3310 NM      DS    1
040F                  3320 SA      DS    3
0412                  3330 FA      DS    3
0415                  3340 CR      EQU   13
0415                  3350 DOS     EQU   2028H
```

SYMBOL TABLE

AHE1	01FE	AHEX	01FB	ASH1	0212	BIN1	01F1	BINH	01DE
BYTE	0402	CHIN	0195	CHOT	01D4	CHOUT	01BB	CR	000D
CRLF	01C3	DOS	2028	ENCK	0160	FA	0412	FIN	017D
HCON	040C	HOUT	01CE	LF	01C8	LK1	0042	LK2	0069
LOOK	0092	LOOK1	0098	LOOK2	00B5	LOOK3	00BA	MA	010F
MA1	0157	NBR	0400	NM	040E	NUM	0401	PRT	00C7
PRT1	00E8	PRT2	0105	REV	0188	SA	040F	ST1	0011
STACK	0400	START	0000	STOUT	021A	ZBU1	022C	ZBUF	0226

A Memory Catalog Program

By Jim Baumgardt

For those of us who are not blessed with 65K of memory in our computers it is often necessary to "move" (readdress) a RAM board to accommodate our software. It would be nice to have a way of quickly checking to see that the RAM we moved did indeed arrive at the desired address space.

With this in mind, I wrote the memory catalog program. In order to satisfy the "readily available" requirement, the program should reside in PROM. The version listed here is written to stand alone in either RAM or PROM. It requires less than a ¼ K of memory and a teletype or CRT terminal as an output device. I included it as a command in my system monitor so that it can be invoked by depressing a single key on my console. This gives me a map of my memory allocations. I have also found that it is a quick and reassuring check of the system when first powering up.

It was assembled using ESP-1 to run at location 0000 but can be reassembled to run anywhere. The output routine is written for a teletype or CRT and is located at the end of the program so that it can be easily modified to suit your system.

The monitor entry point is the location that the program will jump to when it is finished running and should be changed to fit your system. If this program is to be called from another program, then the jump to MON at location 3F HEX should be changed to a RETurn and the LXI stack at location 0000 should be eliminated.

The stack is set by the first instruction and should be set up to your system, if the program is to stand alone.

When running, the program will scan all memory from location 0000 to FFFF HEX and print on the console in HEX notation the starting and ending addresses of all good RAM. It will not permanently alter the contents of the memory and so can be run at any time. The program will also test over its own stack.

The memory is tested by reading a byte, complementing it, and writing it back into the same location. It is then read out again, recomplemented and compared with the byte originally read out. If it compares, it is then stored back in the location.

If it encounters a PROM or a bad location in RAM it will be treated the same as unassigned memory space. Figure 1 is an example of the output on my system. Figure 2 is a second run after moving an 8K block of RAM.□

PROGRAM LISTING

```
0000                *        *** 8080 MEMORY CATALOG PROGRAM ***
0000                *
0000                *
0000                * WRITTEN BY JIM BAUMGARDT           NOVEMBER '77
0000                *
0000                * HARDWARE - ANY 8080 COMPUTER WITH TELETYPE OR
0000                *            CRT DIVICE FOR OUTPUT.
0000                *
0000                * SOFTWARE - NO ADDITIONAL SOFTWARE REQUIRED.
0000                *
0000                * MEMORY ALLOCATION - 9B HEX BYTES OF PROM OR
0000                *                     RAM FOR PROGRAM STORAGE.
0000                *                     8 BYTES OF RAM FOR STACK
0000                *                     (4 LEVELS).
0000                *
0000                *
0000                *
0000                *
0000                STATS   EQU   00H          CONSOLE STATUS PORT
0000                DATA    EQU   01H          DATA PORT
0000                MON     EQU   0F800H       MONITOR ENTRY POINT
0000                *
0000                *
0000                ORG     ORG   0
0000 31 00 04               LXI   SP,400H      SET STACK
0003 06 0D                  MVI   B,0DH        PRINT CR,LF
0005 CD 91 00               CALL  OUT
0008 06 0A                  MVI   B,           0AH
000A CD 91 00               CALL  OUT
000D 21 00 00               LXI   H,0          SET BEGINNING ADD
0010 11 00 00               LXI   D,ORG        SAVE STARTING ADD
0013 CD 83 00       LOOP1   CALL  TEST         TEST FOR GOOD LOCATION
0016 CA 20 00               JZ    CONT         CONTINUE IF GOOD
0019 23                     INX   H            TRY NEXT
001A CD 8C 00               CALL  DONE         FFFF YET?
001D C2 13 00               JNZ   LOOP1        NO
0020 CD 5D 00       CONT    CALL  ADROT        PRINT STARTING ADD
0023 06 2D                  MVI   B,02DH       PRINT '-'
0025 CD 91 00               CALL  OUT
0028 7A             LOOP2   MOV   A,D          GET ORG ADD
0029 BC                     CMP   H            IS THIS PROG AREA
002A C2 2E 00               JNZ   BYPAS        NO
002D 24                     INR   H            BUMP ADD 1 PAGE
002E CD 83 00       BYPAS   CALL  TEST         TEST 1 LOCATION
0031 C2 42 00               JNZ   ERROR        ERROR IF BAD
0034 23                     INX   H            BUMP ADD
0035 CD 8C 00               CALL  DONE         FFFF YET?
0038 C2 28 00               JNZ   LOOP2        NO
003B 2B                     DCX   H            LAST GOOD ADD
003C CD 5D 00               CALL  ADROT        PRINT ENDING ADD
003F C3 5A 00               JMP   RET          ALL DONE
0042 2B             ERROR   DCX   H            LAST GOOD ADD
0043 CD 5D 00               CALL  ADROT        PRINT ENDING ADD
0046 06 20                  MVI   B,20H        PRINT SPACE
0048 CD 91 00               CALL  OUT
004B 23                     INX   H            BUMP ADD
004C 23                     INX   H            AGAIN
004D CD 83 00       LOOP3   CALL  TEST         TEST 1 LOCATION
0050 CA 13 00               JZ    LOOP1        FOUND A GOOD ONE
0053 23                     INX   H            BUMP ADD
0054 CD 8C 00               CALL  DONE         FFFF YET?
0057 C2 4D 00               JNZ   LOOP3        TEST NEXT LOCATION
005A C3 00 F8       RET     JMP   MON          THATS ALL
005D                *
005D                * ADDRESS OUT - PRINTS THE CONTENTS OF THE H&L
005D                *               REGESTERS ON THE CONSOLE.
005D                *
005D 7C             ADROT   MOV   A,H          PRINT FIRST HALF
005E CD 66 00               CALL  HEXOT        OF ADDRESS
0061 7D                     MOV   A,L          PRINT SECOND HALF
0062 CD 66 00               CALL  HEXOT
0065 C9                     RET
0066                *
0066                * HEX OUT - CONVERTS THE HEX VALUE IN A TO TWO
0066                *           ASCII CHARACTERS AND PRINTS THEM.
0066                *
0066 F5             HEXOT   PUSH  PSW          SAVE THE CHARACTER
0067 0F                     RRC                SHIFT UPPER 4 BITS
0068 0F                     RRC
0069 0F                     RRC
006A 0F                     RRC
006B CD 79 00               CALL  HASCI        CONVERT TO ASCII
006E CD 91 00               CALL  OUT          PRINT IT
0071 F1                     POP   PSW          GET CHARACTER
0072 CD 79 00               CALL  HASCI        CONVERT IT
0075 CD 91 00               CALL  OUT          PRINT IT
0078 C9                     RET
0079                *
0079                * HEX TO ASCII - CONVERTS THE HEX VALUE IN A TO
0079                *                AN ASCII CHARACTER.
0079                *
0079 E6 0F          HASCI   ANI   00FH         MASK UPPER 4 BITS
007B C6 90                  ADI   90H          A-F WILL CAUSE CARRY
007D 27                     DAA                MAKE DECIMAL
007E CE 40                  ACI   40H          ACC CARRY
0080 27                     DAA                MAKE DECIMAL
0081 47                     MOV   B,A          SAVE IN B
0082 C9                     RET
0083                *
0083                * TEST - TESTS ONE BYTE OF MEMORY
0083                *        NON-DESTRUCTIVLY.
0083                *
0083 7E             TEST    MOV   A,M          GET DATA
0084 47                     MOV   B,A          SAVE IT
0085 2F                     CMA                COMPLIMENT DATA
0086 77                     MOV   M,A          STORE IT BACK
0087 7E                     MOV   A,M          RETRIVE IT
0088 2F                     CMA                RECOMPLIMENT IT
0089 B8                     CMP   B            IS IT GOOD
008A 77                     MOV   M,A          RESTORE LOCATION
008B C9                     RET
008C                *
008C                * DONE - DETERMINS IF THE H&L ARE = TO ZERO
008C                *
008C AF             DONE    XRA   A            CLEAR A
008D BD                     CMP   L            IS L = 0
008E C0                     RNZ                NO
008F BC                     CMP   H            IS H = 0
0090 C9                     RET                Z FLAG = 1 IF H&L = 0
0091                *
0091                * OUT - OUTPUTS THE CHARACTER IN B TO THE CONSOLE
0091                *
0091 DB 00          OUT     IN    STATS        READ STATUS
0093 E6 80                  ANI   80H          MASK STATUS BIT
0095 C2 91 00               JNZ   OUT          WAIT FOR STATUS
0098 78                     MOV   A,B          GET DATA
0099 D3 01                  OUT   DATA         PRINT IT
009B C9                     RET

SYMBOL TABLE

ADROT 005D   BYPAS 002E   CONT  0020   DATA  0001   DONE  008C   ERROR 0042
HASCI 0079   HEXOT 0066   LOOP1 0013   LOOP2 0028   LOOP3 004D   MON   F800
ORG   0000   OUT   0091   RET   005A   STATS 0000   TEST  0083
```

```
0000-4FFF CC00-CFFF F400-F4FF
0000-4FFF CC00-CFFF F400-F4FF
sample run of MEMORY CATALOG PROGRAM   2 runs
```

Figure 1.

```
0000-1FFF 4000-4FFF 6000-7FFF CC00-CFFF F400-F4FF
0000-1FFF 4000-4FFF 6000-7FFF CC00-CFFF F400-F4FF
Sample run of MEMORY CATALOG PROGRAM   2 runs
```

Figure 2.

DIGI-KEY CORPORATION

Quality Electronic Components

TOLL FREE 1-800-346-5144

MINNESOTA RESIDENTS 218-681-6674

DON'T FORGET OUR DISCOUNTS WHEN COMPARING PRICES

I.C.'S • RESISTORS • TRANSISTORS • CAPACITORS • DIODES • I.C. SOCKETS & PINS • SWITCHES
CLOCK MODULES • OPTOELECTRONICS • BREADBOADING & TESTING DEVICES • DRAFTING SUPPLIES
DATA BOOKS • HEAT SINKS • WIRE • TOOLS . . . AND MORE . . . WRITE FOR FREE CATALOG . . . .

INTEGRATED CIRCUITS

TTL

7400	.21
7401	.21
7402	.21
7403	.21
7404	.21
7405	.21
7406	.25
7407	.25
7408	.21
7409	.21
7410	.21
7411	.21
7412	.21
7413	.25
7414	.89
7416	.25
7417	.25
7420	.21
7421	.25
7423	.35
7425	.35
7426	.25
7427	.33
7428	.28
7430	.21
7432	.25
7433	.30
7437	.25
7438	.25
7439	.50
7440	.21
7442	.53
7445	.70
7446	.70
7447	.70
7448	.70
7450	.21
7451	.21
7453	.21
7454	.21
7460	.21
7470	.30
7472	.30
7473	.30
7474	.30
7475	.49
7476	.32
7480	.70
7483	.70
7485	.89
7486	.28
7489	2.19
7490	.44
7491	.70
7492	.44
7493	.44
7494	.70
7495	.70
7496	.70
74100	1.28
74107	.30
74109	.33
74116	1.26
74121	.35
74122	.44
74123	.61
74125	.40
74126	.40
74128	.48
74132	.70
74141	.88
74145	.70
74147	1.63
74148	1.30
74150	1.16
74151	.70
74153	.65
74154	1.03
74155	.70
74156	.70
74157	.70
74160	.88
74161	.88
74162	.88
74163	.88
74164	.96
74165	1.15
74166	1.15
74170	2.64
74174	.98
74175	.93
74176	.79
74177	.79
74180	.70
74180	.70
74181	2.15
74182	.79
74184	2.15
74185	2.19
74189	3.50
74190	1.23
74191	1.23
74192	.88
74193	.88
74194	.88
74195	.88
74196	.88
74197	.88
74198	1.49
74199	1.49
74221	.86
74251	1.09
74279	.55
74298	.94
74365	.67
74366	.67
74367	.67
74368	.67

74LS00

74LS00	.28
74LS01	.28
74LS02	.28
74LS03	.28
74LS04	.29
74LS05	.29
74LS08	.29
74LS09	.29
74LS10	.28
74LS11	.28
74LS12	.28
74LS13	.47
74LS14	1.02
74LS15	.28
74LS20	.28
74LS21	.28
74LS22	.28
74LS26	.33
74LS27	.33
74LS28	.33
74LS30	.28
74LS32	.33
74LS33	.33
74LS37	.33
74LS38	.33
74LS40	.28
74LS42	.67
74LS47	.79
74LS48	.77
74LS51	.28
74LS54	.28
74LS55	.28
74LS73	.39
74LS74	.39
74LS75	.52
74LS76	.39
74LS78	.39
74LS83	.79
74LS85	.49
74LS86	.39
74LS90	.57
74LS92	.57
74LS93	.57
74LS95	.88
74LS86	.88
74LS107	.39
74LS109	.39
74LS112	.39
74LS113	.39
74LS114	.39
74LS125	.49
74LS126	.49
74LS132	.81
74LS136	.39
74LS138	.73
74LS139	.73
74LS151	.75
74LS153	.75
74LS154	1.10
74LS155	.75
74LS156	.75
74LS157	.75
74LS158	.75
74LS160	1.02
74LS161	1.02
74LS163	1.02
74LS164	1.02
74LS168	1.14
74LS169	1.14
74LS170	1.73
74LS173	1.34
74LS174	1.06
74LS175	.84
74LS181	2.50
74LS190	1.18
74LS191	1.18
74LS192	.98
74LS193	.53
74LS194	.88
74LS195	.86
74LS196	.86
74LS197	.86
74LS197	.86
74LS221	1.06
74LS240	1.90
74LS241	1.90
74LS242	1.70
74LS243	1.70
74LS244	1.70
74LS247	.79
74LS248	.79
74LS249	.79
74LS251	.84
74LS253	.84
74LS257	.75
74LS258	.75
74LS260	.34
74LS261	2.00
74LS266	.39
74LS273	1.28
74LS279	.62
74LS283	.79
74LS283	.65
74LS293	.65
74LS295	1.12
74LS298	1.30
74LS365	.67
74LS366	.67
74LS367	.67
74LS368	.67
74LS377	1.50
74LS386	.39
74LS395	1.74
74LS670	2.34
81LS95	.77
81LS96	.77
81LS97	.77
81LS98	.77

INTERFACE

8093	.40
8094	.40
8095	.67
8096	.67
8097	.67
8098	.67
75450	.88
75451	.61
75452	.61
75453	.61
75454	.61
75491	.81
75492	.84
75493	1.09
75494	1.19
MC1488N	.90
MC1489N	.90

4000 CMOS

4000	.23
4001	.23
4002	.23
4006	1.23
4007	.23
4008	.79
4009	.44
4010	.44
4011	.23
4012	.23
4013	.40
4014	.96
4015	.96
4016	.40
4017	1.05
4018	1.05
4019	.23
4020	1.14
4021	1.14
4022	.96
4023	.23
4024	.84
4025	.23
4026	1.68
4027	.40
4028	.89
4029	1.14
4030	.23
4033	1.51
4034	3.50
4035	1.14
4040	1.14
4041	.79
4042	.79
4043	.70
4044	.70
4046	1.86
4049	.40
4050	.40
4051	1.26
4052	1.26
4053	1.26
4060	1.58
4066	.79
4071	.23
4072	.23
4073	.23
4075	.23
4081	.23
4082	.23
4502	.79
4510	1.14
4511	1.05
4514	2.80
4515	2.80
4516	1.23
4518	1.14
4520	1.14
4527	1.68
4528	.88
4585	1.23
74C00	.23

LINEAR

CA3046	.75
CA3086	.50
LF351N	.76
LF356N	1.20
LF357N	1.20
LF13741	.716
LM301AN	.44
LM307N	.64
LM308N	1.00
LM309K	1.80
LM311N	.90
LM317T	3.00
LM317MP	1.90
LM318N	2.70
LM320MP-5	1.30
LM320MP-6	1.30
LM320MP-8	1.30
LM320MP-9	1.30
LM320MP-12	1.30
LM320MP-15	1.30
LM320MP-18	1.30
LM320MP-24	1.30
LM323K	7.50
LM324N	.72
LM325N	2.70
LM326N	2.70
LM327N	2.70
LM339N	1.15
LM340T-5	1.25
LM340T-6	1.25
LM340T-8	1.25
LM340T-10	1.25
LM340T-12	1.25
LM240T-15	1.25
LM340T-18	1.25
LM340T-24	1.25
LM341P-5	.98
LM341P-6	.98
LM341P-8	.98
LM341P-10	.98
LM341P-12	.98
LM341P-15	.98
LM341P-18	.98
LM341P-24	.98
LM348N	1.60
LM349N	1.60
LM358N	.72
LM3110N	.84
LM1800N	.94
LM1801N	1.70
LM1812N	4.80
LM1830N	1.30
LM1889N	2.00
LM2111N	1.25
LM2907N	1.50
LM2917N	1.50
LM3089N	1.30
LM3900N	.88
LM3900N	.88
LM3905N	1.40
LM3909N	.69
LM3911N	1.50
MC1456V	1.00
MC1458V	.53
MC1496N	.70
MC3302P	1.15
NE536T	2.50
NE540L	2.04
NE555V	.48
NE556A	.88
NE558N	2.40
NE559N	2.40
NE560B	2.50
NE561B	2.50
NE562B	2.50
NE565A	1.25
NE566V	1.28
NE567V	1.36
UA709CV	.44
UA710CA	.44
UA711CA	.53
UA723CA	.60
UA733CA	.75
UA741CV	.44
UA747CA	.70
UA748CV	.49
UA7805CU	1.25
UA7806CU	1.25
UA7808CU	1.25
UA7812CU	1.25
UA7812CU	1.25
UA7815CU	1.25
UA7818CU	1.25
UA7824CU	1.25
UA78L05ACZ	.40
UA78L06ACZ	.40
UA78L08ACZ	.40
UA78L1oACZ	.40
UA78L12ACZ	.40
UA78L15ACZ	.40
UA78L18ACZ	.40
UA78L24ACZ	.40
UA7905CU	1.70
UA7906CU	1.70
UA7908CU	1.70
UA7909CU	1.70
UA7912CU	1.70
UA7915CU	1.70
UA7918CU	1.70
UA7924CU	1.70

MISC IC

MM5314N	3.75

I.C. SOCKETS

8 PIN SOLDER	.17	1.60/10	14.50/C
14 PIN SOLDER	.20	1.90/10	17.00/C
16 PIN SOLDER	.22	2.10/10	18.75/C
18 PIN SOLDER	.29	2.75/10	24.50/C
20 PIN SOLDER	.30	2.85/10	25.25/C
22 PIN SOLDER	.33	3.15/10	27.75/C
24 PIN SOLDER	.38	3.60/10	32.25/C
28 PIN SOLDER	.45	4.25/10	38.25/C
40 PIN SOLDER	.63	6.00/10	53.50/C
8 PIN W-W	.24	2.30/10	20.50/C
14 PIN W-W	.26	2.50/10	22.00/C
16 PIN W-W	.30	2.85/10	25.50/C
18 PIN W-W	.60	5.70/10	51.00/C
20 PIN W-W	.70	6.65/10	59.50/C
22 PIN W-W	.84	8.00/10	71.50/C
24 PIN W-W	.96	9.10/10	81.50/C
28 PIN W-W	1.12	10.00/10	95.20/C
40 PIN W-W	1.36	12.50/10	115.00/C

MOLEX I.C. SOCKET PINS

.85/C 8.20/M 38.20/5M 275.00/50M

SLIDE SWITCHES

SPST	.15	1.25/10	10.00/C
SPDT	.19	1.70/10	13.00/C
DPDT	.23	2.00/10	19.00/C

STRANDED HOOK UP WIRE

20 ga PVC	2.50/100'	10.00/500'
22 ga PVC	2.80/100'	11.25/500'
24 ga PVC	2.10/100'	8.50/500'
26 ga PVC	2.10/100'	8.50/500'

50 VOLT DISCS

100 PF	.40/10	3.50/C
220 PF	.40/10	3.50/C
470 PF	.40/10	3.50/C
.001 UF	.40/10	3.50/C
.0022 UF	.40/10	3.50/C
.0047 UF	.40/10	3.50/C
.01 UF	.45/10	3.65/C
.022 UF	.50/10	4.00/C
.047 UF	.70/10	5.60/C
.1 UF	1.35/10	11.50/C

8080A CHIP SET

ONE EACH 8080A, 8212, 8224 and 8228 PLUS SIXTEEN 2102-1's — ALL FOR

$49.95

MICROPROCESSORS

Best Values Yet!

8080A	8-BIT CPU, 2 USEC CYCLE	$12.95
8212	8-BIT I/O PORT	$ 3.95
8216	BI-DIRECTIONAL BUS DRIVER	$ 4.25
8224	CLOCK GENERATOR	$ 3.95
8228	SYSTEM CONTROLLER	$ 6.95
8251	COMMUNICATIONS INTERFACE	$12.95
8255	PERIPHERAL INTERFACE	$11.95
1702A	2K EPROM (256x8)	$ 6.95
2708	8K EPROM (1Kx8)	$29.50
2102-1	1K RAM 500 NS	$ 1.79
SC/MP	5 VOLT SC/MP CPU	$ 9.95
2650	MICROPROCESSOR 1.25 MH-2 CLOCK 40 DIL	$24.95

WIRE WRAPPING WIRE IN BULK

Red or Black 30 ga. Kynar

100' $2.00 500' $8.50 1000' $15.00

5% CARBON FILM RESISTORS

¼ & ½ WATT SIZE

5 pcs/value 5¢ ea., 100 pcs/value 1.7¢ ea., 1000 pcs/value 1.2¢ ea.

PLESSEY SAMPLER

AN ASSORTMENT OF METALIZED POLYESTER CAPACITORS

300 CAPS 18 VALUES $26.00

METALIZED POLYESTER CAPACITORS

.001	.14	1.15/10	9.00/C	.033	.15	1.25/10	10.00/C
.0022	.14	1.15/10	9.00/C	.047	.15	1.25/10	10.00/C
.0033	.14	1.15/10	9.00/C	.068	.15	1.25/10	10.00/C
.0047	.14	1.15/10	9.00/C	.1	.17	1.35/10	11.00/C
.0068	.14	1.15/10	9.00/C	.22	.23	1.85/10	15.00/C
.01	.14	1.15/10	9.00/C	.33	.30	2.50/10	20.00/C
.022	.15	1.25/10	10.00/C	.47	.36	3.00/10	24.00/C

ELECTROLYTIC CAPACITORS

VALUE	RADIAL LEADS			AXIAL LEADS		
.47/50V	.08	.65/10	5.41/C	.11	.90/10	7.65/C
1/50V	.08	.65/10	5.41/C	.11	.90/10	7.65/C
2.2/50V	.08	.65/10	5.41/C	.12	.90/10	7.82/C
3.3/50V	.08	.65/10	5.41/C	.12	1.00/10	8.31/C
4.7/35V	.08	.65/10	5.41/C	.12	.95/10	7.91/C
4.7/50V	.08	.68/10	5.75/C	.12	1.00/10	8.31/C
10/16V	.08	.65/10	5.41/C	.11	.90/10	7.65/C
10/25V	.08	.65/10	5.66/C	.12	1.00/10	8.31/C
10/35V	.09	.70/10	6.12/C	.13	1.10/10	8.94/C
10/50V	.10	.75/10	6.58/C	.14	1.15/10	9.56/C
22/16V	.08	.67/10	5.66/C	.12	1.00/10	8.31/C
22/25V	.09	.70/10	6.09/C	.13	1.05/10	8.74/C
22/35V	.11	.85/10	7.29/C	.15	1.19/10	9.98/C
22/50V	.12	1.00/10	8.48/C	.17	1.32/10	11.22/C
33/16V	.09	.75/10	6.90/C	.12	1.00/10	8.48/C
33/25V	.10	.81/10	7.50/C	.14	1.15/10	9.56/C
33/35V	.13	1.05/10	9.65/C	.17	1.34/10	11.23/C
33/50V	.14	1.13/10	10.41/C	.19	1.52/10	12.89/C
47/10V	.09	.71/10	6.52/C	.13	1.04/10	9.50/C
47/16V	.10	.81/10	7.47/C	.14	1.15/10	9.56/C
47/25V	.13	1.05/10	9.65/C	.17	1.30/10	11.22/C
47/35V	.14	1.13/10	10.41/C	.19	1.51/10	12.89/C
45/50V	.15	1.21/10	11.16/C	.21	1.71/10	14.55/C
100/10V	.10	.77/10	6.58/C	.14	1.13/10	9.56/C
100/16V	.11	.85/10	7.28/C	.17	1.30/10	11.22/C
100/25V	.13	1.10/10	9.15/C	.20	1.55/10	13.30/C
100/35V	.17	1.41/10	11.85/C	.25	1.93/10	16.50/C
100/50V	.21	1.71/10	14.55/C	.29	2.30/10	19.70/C
220/10V	.13	1.08/10	9.15/C	.18	1.42/10	12.05/C
220/16V	.15	1.16/10	9.86/C	.20	1.55/10	13.30/C
220/25V	.21	1.71/10	14.55/C	.29	2.35/10	19.96/C
220/35V	.25	2.03/10	17.26/C	.35	2.79/10	23.70/C
220/50V	.29	2.35/10	19.96/C	.40	3.23/10	27.44/C
330/6V	.14	1.12/10	9.50/C	.19	1.48/10	13.71/C
330/10V	.15	1.16/10	9.83/C	.21	1.64/10	15.13/C
330/16V	.21	1.66/10	14.14/C	.31	2.45/10	22.70/C
330/25V	.23	1.86/10	15.79/C	.38	3.07/10	28.38/C
330/35V	.33	2.66/10	24.59/C	.43	3.43/10	31.68/C
330/50V	.54	4.30/10	39.73/C	.60	4.81/10	44.45/C
470/6V	.15	1.21/10	11.16/C	.20	1.64/10	15.13/C
470/10V	.21	1.71/10	14.55/C	.31	2.45/10	22.70/C
470/16V	.23	1.81/10	15.39/C	.33	2.66/10	24.59/C
470/25V	.29	2.35/10	19.96/C	.43	3.43/10	31.68/C
470/35V	.41	3.27/10	30.26/C	.47	3.78/10	34.99/C
470/50V	.54	4.30/10	39.73/C	.75	6.03/10	55.80/C
1000/6V	.22	1.90/10	16.50/C	.35	2.76/10	25.54/C
1000/10V	.24	1.96/10	16.62/C	.38	3.07/100	28.38/C
1000/16V	.29	2.35/10	19.96/C	.43	3.43/10	31.68/C
1000/25V	.42	3.33/10	28.27/C	.68	5.42/10	50.13/C
1000/35V	.60	4.81/10	44.45/C	.75	6.03/10	55.81/C
2200/6V	.36	2.86/10	26.48/C	.43	3.43/10	31.68/C
2200/10V	.42	3.33/10	28.27/C	.60	4.81/10	44.45/C
2200/16V	.54	4.30/10	36.58/C	.68	5.42/10	50.13/C

Bishop Graphics

The Digi-Key Catalog features many more quality Bishop Graphics products.

No. 3365 $12.50

JMJ DIGITAL DISPLAY BEZEL

NEW

$4.95

PATENTED CHROMAFILTER SCREEN, DIE CAST METAL FRAME MOUNTS IN PANELS UP TO 3/16" THICK, NO EXPOSED HARDWARE, SCRATCH RESISTANT AND EASILY CLEANED, ELIMINATES GLARE.

MOUNTING ADAPTERS AVAILABLE TO SIMPLIFY MOUNTING AND ASSURE PERFECT ALIGNMENT.

400 BEZEL WITH RED FILTER	4.95
420 MOUNTING ADAPTER FOR MA1001 & MA1002	.69
421 MOUNTING ADAPTER FOR MA1010	.69

CLOCK MODULES

The MA1002 and MA1010 series clock modules by National Semiconductor are fully assembled and tested clocks using a 4 digit LED display and an MOS integrated circuit on the same board. Simply connect switches and our special transformer and you have a fully functioning clock.

The MA1003 clock module is a fully assembled and tested 12 hour clock using a high brilliance green flourescent display and crystal time base making it perfect for car, boat or other portable use. It operates directly from 12 volts DC so no transformer is needed. Our price includes three push button switches for setting the time.

MA1002A .5" LED 12 Hour AM-PM Clock Module	$10.50
MA1002A SET Module with Transformer & Switches	$13.95
MA1002C .5" LED 24 Hour Clock Module	$10.50
MA1002C SET Module w/Transformer & Switches	$13.95
MA1003 12 Volt Car Clock with Switches	$24.95
MA1010A .84" LED 12 Hour AM-PM Clock Module	$13.00
MA1010A SET Module w/Transformer & Switches	$16.45
MA1010C .84" LED 24 Hour Clock Module	$13.00
MA1010C SET Module w/Transformer & Switches	$16.45

MA1002 0.5" High Digits

Actual Size: 1.375"x3.05"

MA1010 0.84" High Digits

Actual size: 1.75"x3.75"

MA1003 $24.95 NEW!

LED DIGITS AND LAMPS

FND357 .375" Single Digit, CC, Red	$1.35
FND500 .5" Single Digit, CC, Red	1.35
FND507 .5" Single Digit, CC, Red	1.35
FND800 .8" Single Digit, CC, Red	2.50
FND807 .8" Single Digit, CA, Red	2.50
NSN373 .3" Dual Digits, CC, Red	2.20
NSN374 .3" Dual Digits, CA, Red	2.20
NSN583 .5" Dual Digits, CC, Red	2.60
NSN584 .5" Dual Digits, CA, Red	2.60
NSN783 .7" Dual Digits, CC, Red	3.00
NSN784 .7" Dual Digits, CA, Red	3.00
NSL5053 Dif, Red T-1¾ Lamp w/Mtg. Hdwe	$1.80/10 $16.00/100
NSL5056 Dif, Narrow Angle Red T-1¾ Lamp w/Mtg. Hdwe	$1.80/10 $16.00/100

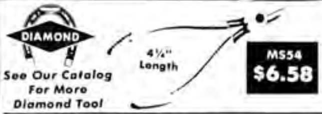
5% CARBON FILM RESISTORS

¼ WATT RESISTOR ASSORTMENTS

Catalog Number	Description	PRICE
RS125	Set of 5 each of the 73 standard 5% ¼ watt carbon film resistors in the series 1.0, 1.2, 1.5, 1.8, 2.2, etc., through 1.0 megohm (365 total pcs.)	$9.90
RS225	Set of 5 each of the 72 standard 5% ¼ watt carbon film resistors in the series 1.1, 1.3, 1.6, 2.0, 2.4, etc., through 910 kilohm (360 total pcs.)	$9.90

½ WATT RESISTOR ASSORTMENTS

Catalog Number	Description	PRICE
RS150	Set of 5 each of the 73 standard 5% ½ watt carbon film resistors in the series 1.0, 1.2, 1.5, 1.8, 2.2, etc., through 1.0 megohm (365 total pcs.)	$9.90
RS250	Set of 5 each of the 72 standard 5% ½ watt carbon film resistors in the series 1.1, 1.3, 1.6, 2.0, 2.4, etc., through 910 kilohm (360 total pcs.)	$9.90



MS54 $6.58

See Our Catalog For More Diamond Tool

WSU-30 Regular Wrap	$5.95
WSU-30M Modified Wrap	$6.95

STRIPS — WRAPS — UNWRAPS

SILICON TRANSISTORS

MPS918	.16	1.55/10	13.60/C	130.00/M
MPS930	.16	1.55/10	13.60/C	130.00/M
MPS2222A	.16	1.55/10	13.60/C	130.00/M
MPS2369A	.16	1.55/10	13.60/C	130.00/M
MPS2907A	.16	1.55/10	13.60/C	130.00/M
MPS3393	.16	1.55/10	13.60/C	130.00/M
MPS3393	.16	1.55/10	13.60/C	130.00/M
MPS3394	.16	1.55/10	13.60/C	130.00/M
MPS3395	.16	1.55/10	13.60/C	130.00/M
MPS3563	.16	1.55/10	13.60/C	130.00/M
MPS3565	.16	1.55/10	13.60/C	130.00/M
MPS3638A	.16	1.55/10	13.60/C	130.00/M
MPS3640	.16	1.55/10	13.60/C	130.00/M
MPS3641	.16	1.55/10	13.60/C	130.00/M
MPS3643	.16	1.55/10	13.60/C	130.00/M
2N3904	.16	1.55/10	13.60/C	130.00/M
2N3906	.16	1.55/10	13.60/C	130.00/M
2N4124	.16	1.55/10	13.60/C	130.00/M
2N4126	.16	1.55/10	13.60/C	130.00/M
2N4401	.16	1.55/10	13.60/C	130.00/M
2N4403	.16	1.55/10	13.60/C	130.00/M
2N4410	.16	1.55/10	13.60/C	130.00/M
2N5087	.16	1.55/10	13.60/C	130.00/M
2N5089	.16	1.55/10	13.60/C	130.00/M
2N5210	.16	1.55/10	13.60/C	130.00/M
2N3055	.99	9.20/10	85.00/C	800.00/M
MPF102	.36	3.35/10	30.60/C	300.00/M
2N5457	.48	4.50/10	41.00/C	400.00/M
MPSA13	.28	2.60/10	24.00/C	230.00/M
MJ2955	.99	9.20/10	85.00/C	800.00/M
TIP120	.99	9.20/10	85.00/C	800.00/M

½ WATT ZENER DIODES

.15 1.30/10 11.00/C

1N5226B	3.3V	1N5242B	12V
1N5227B	3.6V	1N5243B	13V
1N5228B	3.9V	1N5244B	14V
1N5229B	4.3V	1N5245B	15V
1N5230B	4.7V	1N5246B	16V
1N5231B	5.1V	1N5247B	17V
1N5232B	5.6V	1N5248B	18V
1N5233B	6.0V	1N5249B	19V
1N5234B	6.2V	1N5250B	20V
1N5235B	6.8V	1N5251B	22V
1N5236B	7.5V	1N5252B	24V
1N5237B	8.2V	1N5253B	25V
1N5238B	8.7V	1N5254B	27V
1N5239B	9.1V	1N5255B	28V
1N5230B	10V	1N5256B	30V
1N5241B	11V	1N5257B	33V

HARDWARE

2-56 1/4 SCREW	.99/C
2-56 1/2 SCREW	.99/C
4-40 1/4 SCREW	.55/C
4-40 1/2 SCREW	.60/C
6-32 1/4 SCREW	.65/C
6-32 1/2 SCREW	.75/C
8-32 3/8 SCREW	.90/C
8-32 5/8 SCREW	.99/C
2-56 HEX NUT	.55/C
4-40 HEX NUT	.55/C
6-32 HEX NUT	.60/C
8-32 HEX NUT	.60/C
NO 2 LOCKWASHER	.85/C
NO 4 LOCKWASHER	.45/C
NO 6 LOCKWASHER	.45/C
NO 8 LOCKWASHER	.45/C

SILICON DIODES

1N4001	.64/10	5.50/C	$49/M
1N4002	.66/10	5.60/C	$51/M
1N4003	.68/10	5.80/C	$52/M
1N4004	.70/10	5.95/C	$54/M
1N4005	.82/10	7.05/C	$63/M
1N4006	.90/10	7.75/C	$69/M
1N4007	.99/10	8.60/C	$77/M
1N4148	.40/10	3.50/C	$29/M

NEW

FREE CATALOG

DATA BOOKS

TTL I.C.'s 592 P.	$4.00
LINEAR I.C.'s 952 P.	$5.00
CMOS 74C 555 P.	$3.00
POWER TRANSISTORS 151 P.	$3.00
MEMORY I.C.'s 592 P.	$3.00
MOS/LSI I.C.'s 713 P.	$4.00
INTERFACE I.C.'s 464 P.	$4.00
VOLTAGE REG I.C.'s 128 P.	$3.00
LINEAR AP I 432 P.	$4.00
LINEAR AP II 246 P.	$3.00
AUDIO HANDBOOK 196 P.	$3.00
SPECIAL FUNCTION	$3.00
SC/MP APPLICATIONS	$5.00
8080A DESIGN	$5.00
FET DATABOOK	$3.00
CMOS 4000 278 P.	$3.00
ANALOG MANUAL 637 P.	$5.95

DOUBLE DIGIT DISCOUNTS SAVE YOU EVEN MORE!

HANDLING CHARGES

$ 0.00-$ 4.99	Add $2.00
$ 5.00-$24.99	Add $0.75
$ 25.00-$49.99	Add $0.50
$ 50.00-$99.99	Add $0.25
$100.00 & Up	No Charge

VOLUME DISCOUNT & HANDLING CHARGE

All items, except Bishop Graphics and Economy Terminals may be combined for volume discount. First—total all discountable items and apply the volume discount. Then add to this subtotal the nondiscountable items. Add the handling charge. This is your total cost. We pay all shipping and insurance to addresses in the U.S.A. and Canada.

VOLUME DISCOUNT

$ 0.00-$ 24.99	NET
$ 25.00-$ 99.99	Less 10%
$ 100.00-$499.99	Less 15%
$ 500.00-$999.99	Less 20%
$1000.00 & Up	Less 25%

Orders Accepted by Phone or Mail

MasterCharge • Bank-Americard • COD • Check • Money Order

P.O. Box 677, Thief River Falls, Minn. 56701 (218) 681-6674

DIGI-KEY CORPORATION

Quality Electronic Components

P.O. Box 677 Thief River Falls, MN 56701 (218) 681-6674

MICRO-MARKET

IT'S ABOUT TIME!

Finally, a COMPLETE disk utility package for the NORTH STAR MICRO DISK SYSTEM. A MUST for both BUSINESS and hobby systems. PKGUT1 on diskette includes the following four 8080 machine language programs origined at 0 (relocations done no charge): PACKIT — Packs & Unpacks disk files so you can get more storage per disk! CHANGIT — Prints, Dumps and/or Changes data in disk files up to a global level. DORTIT — A generalized sorting utility. COMPIT — File comparison utility. Will compare disk files sequentially or by key and display differences. Diskette with full user's documentation $80.00

North Star BASIC games: ROADRACE, EVENWINS, BIORHYTHM, 3D TICTAC, and the addicting SUPER-WUMP! PKGN1 (5 games on diskette) $15.00

VDM-GAMES (requires SOLOS or CUTER): Real time ROBOTS and ASTEROID! PKGV1 (2 games on 1200 baud CUTS tape) $20.00

Micro Logistics P.O. Box 922, Madison Square Station, New York, NY 10010
N.Y. res. add applicable sales tax

6800 SOFTWARE

INTERRUPT DRIVEN PRINTER ROUTINE — Any BASIC program which uses a printer through a parallel I/O port will run up to 50% faster! $7.50
GRAPH #2 — Create real-time graphics with ASCII characters on a SWTPC GT-61 display with BASIC! Use standard BASIC statements; no need for tedious machine language programming $5.50
The above 2 programs require SWTPC 8K BASIC 2.0.
LIFE—Play J.H. Conway's game of LIFE. Enter a pattern and watch it evolve! $7.50
POTSHOT—Ballistics trajectory game with a realistic moving display. Every play different! (Requires SWTPC GT-61 display) $7.50
Each program requires a 6800 with MIKBUG and less than 2K of memory. Full source listing and MIKBUG paper tape included, or add $1.00 for Kansas City format cassette.

—Write for flyer on our other programs—

APPLIED MICROCOMPUTER SYSTEMS
Box 68, Silver Lake, NH 03875

PROGRAMMING CONTEST!

-WIN-

1st PRIZE - FLOPPY DISK SYSTEM OR $1000 CASH
2nd PRIZE - FLOATING POINT BOARD OR $400 CASH
3rd PRIZE - AUDIO INTERFACE OR $200 CASH

$		M	A	Y
		O		O
W	I	N	$	U
I		E		
N		Y	E	S

OF COURSE YOU HAVE THE SEARCHING AND SORTING SKILLS ACQUIRED BY EVERY HACK PROGRAMMER SINCE ENIAC! BUT DO YOU HAVE A CUNNING AND CLEVERNESS THAT RANKS YOU WITH CIA CALIBRE CODE CRACKERS? PUT YOUR PERSONAL COMPUTER TO WORK AROUND THE CLOCK SOLVING OUR ARDUOUS WORD PUZZLE IN AN ATTEMPT TO:
*** EXPAND YOUR SYSTEM BY WINNING PRIZES ***

For entry form and official rules send a self-addressed stamped envelope to:
MICRO-PUZZLE Dept. B 7858 Cantaloupe Ave. Van Nuys, California 91402

D.C. METRO AREA

TYSONS CORNER, VIRGINIA
COMPUTER SYSTEMS STORE
Microcomputers & Peripherals
for Home, School, & Small Business
1984 Chain Bridge Rd., McLean, VA. 22101
703-821-8333

DIGITAL GROUP OWNERS

10 MAXI- Game Z Cass. 10-MINI
CAT. 1.00
REFUND W/ORD I.A.T.S.Z DEPT 1A
BOX 37 MATTESON, IL 60443

COMPUTER HOBBYISTS!

Nationwide Classified Advertising Newsletter
Buy & Sell Hardware & Software
new/used/unique/low-cost
$3.75 for 18 issues.
Free sample issue on request.
ON_LINE Newsletter
24695 Santa Cruz Hwy, Los Gatos, Ca. 95030

SHORTY CASSETTES

3M C-10 Low Noise High Denisty. Short enough to rewind and still hold many programs.
$1.50 ea. — 10 for $14.00 — Send $1.00 for complete supply catalog with refund coupon.

EA ELLIAM ASSOCIATES
24000 BESSEMER
WOODLAND HILLS, CA. 91367

SOUTH FLORIDA

Sunny Computer Stores, Inc.

- ALPHA MICRO SYSTEMS, MICRO FILE, IMSAI
- BOOKS, MAGAZINES, NEWSPAPERS
- FULL SERVICE
- SOFTWARE
- WRITE OR PHONE US FOR PRICES MORE THAN JUST TOYS

University Shopping Center
1238A South Dixie Highway
Coral Gables, Florida 33146
(305) 661-6042

Check our prices — We will not be undersold!

POLY-8813	1 drive-No monitor	$2500
POLY-88	System 2	$575
IMSAI I-8080	22 Slot	$575
IMSAI	PC Boards/Kits	15% Off
APPLE II	16K	15% Off
EXTENSYS	Memory	20% Off

AAAA Computer How's
1477 Barrington, Suite 17
W. Los Angeles, CA 90025 (213) 477-8478

2708 & 2716 E-PROM PROGRAMMER WITH SOFTWARE

For KIM-1, Motorola 6800, Fairchild F-8, RCA 1802, or Intel 8080.
Assembled & tested, **$59.95.** Kit **$49.95**

Optimal Technology, Inc.
Blue Wood 127, Earlysville, VA 22936

Turn-Key Micro-Computer Systems for professional, commercial and industrial applications. Demonstrations and consultation by appointment. Featuring IMSAI, Applied Data Communication, CP/M and Micro-Soft BASIC.

COMPUTER EMPORIUM
17931 "J" Sky Park Circle, Irvine, CA 92714
(714) 540-8445

PROTECT VALUABLE EQUIPMENT

Lightning and heavy machinery often create AC power line surges. Extensive equipment damage may result. Suppressor absorbs surges, protecting delicate equipment. Sh-h-h!! Built-in filter greatly reduces power-line hash for "whisper" operation. Used in processor, teletype, power supply, peripherals.

ELF/T-10	10 amp Supressor/Filter	$17.25 ppd.
ELF/T-3	3 amp Supressor/Filter	$14.25 ppd.

ELECTRONIC SPECIALISTS, INC., Dept. 1F
171 S. Main St., Natick, Mass. 01760

NORTH STAR SOFTWARE VARIETY PACKAGE #1

- INVENTORY #1
- HEX, DEC. TRANSLATOR AND MEMORY FILL
- ELECTRONIC DESIGN
- MEMORY TEST
- BASIC AND MACHINE CODE PROGRAMS
- GAMES
- MUCH MORE

ALL ON ONE DOUBLE-SIDED DISKETTE ONLY $20 FOR COMPLETE PACKAGE

- ADDITIONAL DOUBLE-SIDED DISKETTES — $5

MC OR VISA WELCOME 24 HOURS

MIDWEST DIGITAL
863 Wood Ave., Wichita, KS 67212
(316) 722-1711

KIM-1 $245

SPECIAL — includes power supply
Immediate Delivery
KIM 4-part harmony music pkg $35
4.5 amp KL Power Supply +5V, +12V $34
Write for list of KIM memory and accessories

A B Computers
P.O. Box 104, Perkasie, PA 18944

MICRO-MARKET AD RATE: $50.00 per column inch. Maximum of 4 column inches per ad. Submit ads to **Micro-Market Ads INTERFACE AGE Magazine,** PO Box 1234, Cerritos, CA 90701.

FIFO FLEA MARKET

FOR SALE: Ohio Scientific OSI 430B Super I/O board, fully populated; A to D, D to A, Multiplex, cassette interface (1200 baud KC), serial I/O, parallel I/O, etc. Brand new. All IC's in sockets. $275. Joe Ruby, 6057 Beck Ave., N. Hollywood, CA 91606, (213) 985-1519.

FOR SALE: Apple II computer with 16K of memory, 2 months old, $1200 or best offer. Also brand new Sharp 13-inch color monitor/TV, $420 firm. Bob Majanski, 214 Coolidge Ave., Hasbrouck Hts., NJ 07604, (201) 288-3742 after 7 p.m.

WANTED: Prom programmer for 5204's. Tom Cage, Box 402, Amarillo, TX 79105.

FOR SALE: Olivetti TE-318, ASR 33 compatible, very quiet, RS-232, 72 characters per line, half/full duplex, only 120 hours of operation. Asking $650. Prapote S., 400 Central Park West, No. 15, New York, NY 10025, (212) 866-5144.

FOR SALE: 8-4115 RAM chips (for S.D. Sales RAM board) $100. Cromemco 16KZ-W 16K RAM board, factory assembled and tested, $495. Sam Lipson, 8 Carlisle Terr., Natick, MA 01760.

FOR SALE: Apple II Software - cassette: 16K Blackjack, multicolor card display, 2-player paddle input, full Las Vegas rules, optional auto-play by computer, sound effects, documentation incl., $10. George W. Lee, 18803 S. Christina Ave., Cerritos, CA 90701.

WANTED: Complete documentation for Viatron Model 5002 9-track computer tape recorder. Must include schematics and parts list for unit and Model 129000-00 power supply. R. M. Schectman, 14606 Larchmere Blvd., Shaker Hts., OH 44120.

MOTOROLA 6800 users. Have nifty trace program that prints out the results of each step: CC, B, A, X, plus the PC, opcode, and instruction of the next step. Program locates anywhere in memory. A beautiful debugging aid. Uses combined hardware/software trace and will trace through ROM. Works as is on any D2 kit interfaced to video or TTY. Works on any MIKBUG or other system with NMI vector user available, with the addition of 2 IC's. Send $3 and SASE for complete listing and instructions. Specify D2/MIKBUG/Other. Lynn Kurtz, 2104 Campo Alegre Rd., Tempe, AZ 85281.

WANTED: Programs in BASIC and BASIC Plus in areas of real estate finance and marketing. W. Blackman, 2741 Beachwood Dr., Hollywood, CA 90068.

FOR SALE: SWTP CT 1024 terminal, 64 characters, scrolling, in custom case. Comes with extra keyboard. $150. Also Tempo FMH 2 meter, handheld, $150. Interested parties send money order or cashier's check for any or all to: Anne Warren, 2980 W. 235th, Apt. 12, Torrance, CA 90505.

FOR SALE OR TRADE: HAL DKB-2010 Morse and Baudot keyboard with memory and manual. $325 or trade for memory, etc. for KIM-1. V.A. Jupe, Star Rte. Box 60, Carlotta, CA 95528, (707) 768-3833.

PAYROLL SOFTWARE: Written in MITS 4.1 BASIC. These programs allow up to 400 different employees on a one-disk system. Complete records are kept on each employee in each pay period. This allows any type of report to be produced. Complete listings $16.00. Listing and diskette $40.00. Call or write Jim Hudson, P.O. Box 4492, Huntsville, AL 35801, (205) 881-7380.

NEW POLY SOFTWARE: Using A00 or A01 BASIC. Over 50 programs to choose from. Send SASE for a complete list. John McCalpin, 2203 Ridge Crest, Richardson, TX 75080.

FOR SALE: Electronic Systems Television typewriter. Assembled, with case, power supply, and keyboard. 1200 baud, parallel or composite video. $100. Assembled board only $50.00. Unit less keyboard $75.00. Vance Navarrette, 2057 S.W. Leewood Ln., Aloha, OR 97005.

FOR SALE: IMSAI, six 4K RAM boards, $100 each; 1 S102-2 board, $150; 1 MIO board, $200; 1 iCOM minifloppy w/controller, FDOS and BASIC-M on diskette, 5 empty diskettes, $825. All are assembled and tested and less than 1 year old. A. Ronen, 649 S. Prospect Ave., Redondo Beach, CA 90207, (213) 372-1969.

FOR SALE: PORTACOM Briefcase ASCII printing terminal w/modem, keyboard, ins. $595. DEC PDP8e's, modules, Ham gear. Free lists. Also buy, trade, repair, design. K2DCY, 11 Squire Hill, N. Caldwell, NJ 07006.

MULTIPLE REGRESSION on a micro? Sure; and if you're interested write: Daniel Buskirk, 1230 York Ave., New York, NY 10021.

FOR SALE: Complete microcomputer system; new (purchased 12/77); fully assembled Byt 8; Digicomm Byteuser (prom); Cromemco D+7A I/O (7 channel analog interface); Dazzler (color TV interface); Byte MPU (processor); SD Sales 4K RAM; all manuals. Cost $1300, best offer takes it, D.G. Linville, 4865 Pasadena Ave., Sacto, CA 95841, (916) 485-4050.

FOR SALE: 3 months old, MITS Altair 8800A, includes 32K RAM, ACR, 2SIO, CPU, ADM-3A, manuals Ext. BASIC on cassette, etc. $3,400 or best offer, plus free tape cassette recorder and 25 game cassettes. First cashier's check or money order to M. Wedgle, 1090 S. Parker Rd., No. A216, Denver, CO 80231 by May 30.

FOR SALE: SWTPC PR-40 printer. Less than one year new. Used very little. Assembled by professional. $190 or best offer. Includes complete documentation and new ribbon. I pay shipping. G. Welker, 6956 Deveron Ridge Rd., Canoga Park, CA 91307.

FOR SALE: Floppy Tape Peripheral. Complete with I/O board, 8 tapes, new and used. Uses stereo 8-track cartridge. Each cartridge can hold one program per track or 8 per cartridge. Program length limited only by tape length. The used tapes have 5 or 6 programs on each and are included free. They are ready to run and cover Star-Trek, Othello, other games and technical programs. Will interface to most microcomputers. Schematic included. Will ship UPS. $100. R. Mendelson, 27 Somerst Pl., Murray Hill, NJ 07974, (201) 464-5244.

CALL FOR ARTICLES

INTERFACE AGE is currently seeking articles covering microcomputer applications in science and medicine. Specifically, we are seeking actual application stories rather than theory. If you have a good idea and are not sure how to put it on paper, check our March 1978 issue for the Style Guide on page 32.

Articles authored by individuals during leisure time are remunerated at a rate from $15.00 to $50.00 per published page and articles describing company projects carry author and company byline, but no honorarium is offered. Articles accepted will be acknowledged with a binder check within 30 days of receipt.

Manuscripts should be double-spaced, typewritten pages, one inch margins, and not less than 3½ pages in length (one published page). Pages should be numbered to insure correct text. Photographs should be numbered and labeled on the backside with a description. Photos should be taken with uniform lighting and background, in the form of glossy black and white prints. Tables, listings, etc., shall be on separate sheets. Computer listings shall be printed using a new ribbon to assure darkest print copy. Authors shall supply a statement of their background, expertise and level of accomplishment.

The publisher assumes no responsibility for artwork, photos, models, or manuscripts. Manuscripts are not acknowledged or returned unless accompanied by a self-addressed, stamped, return envelope.

For article submittal or further information, contact respective editor, INTERFACE AGE Magazine, 16704 Marquardt Avenue, Cerritos, CA 90701 or call (213) 926-9544.

ADVERTISER INDEX

*Manufacturer requests factory-direct inquiry.

Visit Your Nearest MITS® Dealer

ARIZONA
Altair Computer Center
4941 E. 29th St.
Tucson, Arizona 85711
(602) 748-7363

Altair Computer Center
3815 North Third St.
Phoenix, Arizona 85012
(602) 266-1141

ARKANSAS
JFK Electronics
3702 JFK Blvd.
N. Little Rock, Arkansas 72116
(501) 753-1414

CALIFORNIA
Computer Kits
1044 University Ave.
Berkeley, Calif. 94710
(415) 845-5300

COLORADO
Sound-Tronix
900 Ninth Ave.
Greeley, Colorado 80631
(303) 353-1588

Sound-Tronix
3271 Dillon Dr. Pueblo Mall
Pueblo, Colorado 81008
(303) 545-1097

Sound-Tronix
215 Foothills Pkwy.
Foothills Fashion Mall
Fort Collins, Colorado 80521
(303) 221-1700

FLORIDA
Altair Computer Center of Miami
7208 N.W. 56th St.
Miami, Fla. 33166
(305) 887-7408

Altair Computer Center of Orlando
6220 S. Orange Blossom Trail
Suite 602
Orlando, Florida 32809
(305) 851-0913

GEORGIA
The Computer Systemcenter
3330 Piedmont Rd. N.E.
Atlanta, Ga. 30305
(404) 231-1691

ILLINOIS
Chicago Computer Store
517 Talcott Rd.
Park Ridge, Illinois 60068
(312) 823-2388

Chicago Computer Store
919 "B" N. Sheridan Rd.
Peoria, Illinois 61614
(309) 692-7704

Chicago Computer Store
1 Illinois Center Concourse
111 E. Wacker Dr.
Chicago, Illinois 60601

KANSAS
Advanced Micro Systems Inc.
5209 W. 94 Terrace
Prairie Village, Kansas 66207
(913) 648-0600

WEST VIRGINIA AND KENTUCKY
The Computer Store
Suite 5
Municipal Pkg. Bldg.
Charleston, W. Virginia 25301
(304) 345-1360

MASSACHUSETTS
Mits Computer Center
36 Cambridge St.
Burlington, Mass.01803
(617) 272-1162

MICHIGAN
Computer Store of Detroit
505-507 West 11 Mile Rd.
Madison Heights, Michigan 48071
(313) 545-2225

The Computer Store of Ann Arbor
310 E. Washington St.
Ann Arbor, Michigan 48104
(313) 995-7616

MINNESOTA
The Computer Room
3938 Beau D'Rue Dr.
Eagan, Minn. 55122
(612) 452-2567

NEBRASKA
Altair Computer Center
611 North 27th St. #9
Lincoln, Nebraska 68503
(402) 474-2800

NEW MEXICO
Computer Shack
3120 San Mateo N.E.
Albuquerque, New Mexico 87110
(505) 883-8282

NEW YORK
The Computer Store of New York
55 West 39th St.
New York, New York 10018
(212) 221-1404

Micro Systems Store, Inc.
269 Osborne Rd.
Albany, New York 12211
(518) 459-6140

Simplified Business Methods
19 Rector St.
New York, New York 10006
(212) 943-4130

NORTH CAROLINA
Computer Stores of Carolina
1808 E. Independence Blvd.
Charlotte, N.C. 28205
(704) 334-0242

OHIO
Altair Computer Center
5252 North Dixie Drive
Dayton, Ohio 45414
(513) 274-1149

Altair Computer Center
26715 Brook Park Extension
No. Olmsted, Ohio 44070
(216) 734-6266

The Computer Store of Toledo
8 Hillwyck St.
Toledo, Ohio 43615

OKLAHOMA
Altair Computer Center
110 The Annex
5345 East 41st St.
Tulsa, Oklahoma 74135
(918) 664-4564

OREGON
Altair Computer Center
8105 S.W. Nimbus Ave.
Beaverton, Oregon 97005
(503) 644-2314

PENNSYLVANIA
Microcomputer Systems, Inc.
243 West Chocolate Rd.
Hershey, Pa. 17033
(717) 533-5880

TEXAS
Swift Computers, Inc.
3208 Beltline Rd.
Suite 206
Dallas, Texas 75234
(214) 241-4088

Swift Computers, Suite 145
6333 Camp Bowie Blvd.
Ft. Worth, Texas 76116

Altair Computer Center
7302 Harwin Dr.
Suite 206
Houston, Texas 77036
(713) 780-8981

Altair Computer Center
3206-A 34th St.
Lubbock, Texas 79410

UTAH
Microcosm Inc.
534 West 9460 South
South Sandy, Utah 84070
(801) 566-1322

VIRGINIA
Computer-To-Go
1905 Westmoreland St.
Richmond, Va. 23230
(804) 355-5773

Megabyte Computer Assoc.
700 Stoney Point, Suite 7
Newtown Rd.
Norfolk, Va. 23502
(804) 461-3079

Microsystems Computer Corp.
Century Mall—Crystal City
2341 S. Jefferson Davis Hghwy.
Arlington, Va. 22202
(703) 979-5566

WASHINGTON
Pasco Computer Store
6704 Argent Rd.
Pasco, Washington 99301
(509) 547-9014

Altair Computer Center
14100 N.E. 20th St.
Bellevue, Wash. 98007
(206) 641-8800

WISCONSIN
Chicago Computer Store
285 West Northland Ave.
Appleton, Wisconsin 54911
(414) 731-9559

CANADA
The Computer Place
186 Queen St. West
Toronto, Ont. M5V 121
(416) 548-0262
Telex 0622 634

READER SERVICE CARD 2

INTERFACE AGE™

May 1978 Issue
Void after Aug. 31, 1978
(Please type or print)

Name ______________________

Title ______________________

Company ______________ Phone (A/C)______________

Address ______________________ ☐ Home ☐ Business

City ______________ State______ Country______ Zip______

ANSWER THE FOLLOWING BY CHECKING *ONE* BOX ONLY PER QUESTION.

1. I Am A
A. ☐ Professional (Medical, Accounting, Law, Etc.)
B. ☐ Engineer (Electronics, Mechanical Etc.)
C. ☐ Business Person (Retail, Wholesale, Etc.)
D. ☐ Educator (Professor, Teacher, Assistant, Etc.)
E. ☐ Student
F. ☐ Hobbyist
G. ☐ Other ______

2. My Status Is
A. ☐ Have All Computing Equipment
B. ☐ Need More Peripherals
C. ☐ Have CPU Only
D. ☐ Have No Equipment
E. ☐ Other

3. I Need This Information For
A. ☐ Immediate Purchase
B. ☐ Purchase 30-60 Days
C. ☐ Comparisons
D. ☐ Literature Library
E. ☐ Other ______

4. My Interest Emphasis Is
A. ☐ Hardware
B. ☐ Firmware
C. ☐ Software
D. ☐ Other ______

5. My Application Is
A. ☐ Hobby Only
B. ☐ Business Only
C. ☐ Hobby & Business
D. ☐ Instruction Purposes
E. ☐ Research
F. ☐ Design & Development
G. ☐ Other ______

6. My *Primary* Source Of "State-of-the-Art" Information Comes From:
A. ☐ Magazines
B. ☐ Exhibits & Conventions
C. ☐ Club Meetings
D. ☐ Direct Mail From Manufacturers
E. ☐ Other ______

7. I prefer To Buy
A. ☐ Directly From Manufacturers
B. ☐ Local Computer Retailer (Store)
C. ☐ Mail Order
D. ☐ Club Group Purchases
E. ☐ Other ______

8. I Look To *INTERFACE AGE* First For
A. ☐ New Product Information
B. ☐ Software Information
C. ☐ Tutorials
D. ☐ Hardware Articles
E. ☐ Product Advertising
F. ☐ Remarks ______

Please send information on items circled below.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250
251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275
276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325
326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350
351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375
376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400

Please use only 1 card per person. Thanks!

READER SERVICE CARD 1

INTERFACE AGE™

May 1978 Issue
Void after Aug. 31, 1978
(Please type or print)

Name ______________________

Title ______________________

Company ______________ Phone (A/C)______________

Address ______________________ ☐ Home ☐ Business

City ______________ State______ Country______ Zip______

ANSWER THE FOLLOWING BY CHECKING *ONE* BOX ONLY PER QUESTION.

1. I Am A
A. ☐ Professional (Medical, Accounting, Law, Etc.)
B. ☐ Engineer (Electronics, Mechanical Etc.)
C. ☐ Business Person (Retail, Wholesale, Etc.)
D. ☐ Educator (Professor, Teacher, Assistant, Etc.)
E. ☐ Student
F. ☐ Hobbyist
G. ☐ Other ______

2. My Status Is
A. ☐ Have All Computing Equipment
B. ☐ Need More Peripherals
C. ☐ Have CPU Only
D. ☐ Have No Equipment
E. ☐ Other

3. I Need This Information For
A. ☐ Immediate Purchase
B. ☐ Purchase 30-60 Days
C. ☐ Comparisons
D. ☐ Literature Library
E. ☐ Other ______

4. My Interest Emphasis Is
A. ☐ Hardware
B. ☐ Firmware
C. ☐ Software
D. ☐ Other ______

5. My Application Is
A. ☐ Hobby Only
B. ☐ Business Only
C. ☐ Hobby & Business
D. ☐ Instruction Purposes
E. ☐ Research
F. ☐ Design & Development
G. ☐ Other ______

6. My *Primary* Source Of "State-of-the-Art" Information Comes From:
A. ☐ Magazines
B. ☐ Exhibits & Conventions
C. ☐ Club Meetings
D. ☐ Direct Mail From Manufacturers.
E. ☐ Other ______

7. I prefer To Buy
A. ☐ Directly From Manufacturers
B. ☐ Local Computer Retailer (Store)
C. ☐ Mail Order
D. ☐ Club Group Purchases
E. ☐ Other ______

8. I Look To *INTERFACE AGE* First For
A. ☐ New Product Information
B. ☐ Software Information
C. ☐ Tutorials
D. ☐ Hardware Articles
E. ☐ Product Advertising
F. ☐ Remarks ______

Please send information on items circled below.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250
251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275
276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325
326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350
351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375
376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400

Please use only 1 card per person. Thanks!

READER SERVICE CARD

SUBSCRIPTION

SUBSCRIPTION APPLICATION FORM

PAYMENT MUST ACCOMPANY FORM

SUBSCRIPTION FEE per year: $14.00 U.S., $16.00 Canada/Mexico, $24.00 Foreign.
Must be in U.S. funds drawn on a U.S. bank.

Make check payable to: **INTERFACE AGE™ Magazine**
P.O. Box 1234, Cerritos, CA 90701

Name ______________ Title ______________

Company ______________________

Home Address ______________________

City ______________ State ______ Country______ Zip ______

Date ______________ Signature ______________

☐ CHECK ☐ MONEY ORDER ☐ BANKAMERICARD ☐ master charge

Account # ______________ Expires ______________

PLANNING TO MOVE?

Let us know 8 weeks in advance so that you won't miss a single issue of/INTERFACE.

Attach old label where indicated and print new address in space provided. Also include your mailing label whenever you write concerning your subscription. It helps us serve you promptly.

If you have no label handy, print OLD address here.

Please Print

Name

Company

Address

City State Zip code

Affix Label **Print NEW address here**

Name

Company

Address

City State Zip code

MAIL COMPLETED CARD TO:

INTERFACE AGE™ Magazine
P.O. Box 1234
Cerritos, California 90701

MOVING

First Class
Permit No. 217
Clinton, Iowa

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in the United States

Postage will be paid by

INTERFACE AGE™ Magazine

Box 2654

Clinton, Iowa 52732

First Class
Permit No. 217
Clinton, Iowa

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in the United States

Postage will be paid by

INTERFACE AGE™ Magazine

Box 2654

Clinton, Iowa 52732

MORE THAN A HOBBY WITH US.

New iCOM® Attaché™

A serious small system for practical computing

Tinker-time is over for many small-computer users. You're ready for more powerful, more professional equipment—to replace early kluged systems, or for easier computer ownership.

We're ready too. With Attaché—handsomely produced by iCOM® pros, strengthened by our Pertec Computer Corporation family ties. Already first in Microperipherals,® we've expanded our industry commitment, to provide *complete turnkey systems* backed by PCC's full technical resources.

Take Attaché seriously.
At last a no-nonsense desktop computer: highly advanced, but you don't have to be an engineer to run it. And flexible enough for application versatility.

Heavy-duty powered, Attaché's S-100 bus mainframe offers 10-slot accommodations. For 8080 microprocessor, video, and turnkey monitor boards, for full 64-character ASCII keyboard. With room to spare for user options: plug in 16K memory, cassette interface (KCACR), I/O, process control, and/or iCOM floppy disk interface cards.

With serious software support.
Attaché software options take advantage of iCOM's solid experience: field-proven FDOS-III™ Disk Operating System; relocatable assembly language development system; and DEBBI,™ the Disk-Extended BASIC by iCOM.

iCOM: right, from the start.
Now the most-trusted name in Microperipherals can satisfy your *total* system requirements. With a serious computer worth looking into today at your local computer store.

icom®

CIRCLE INQUIRY NO. 28

Products of PCC Pertec Computer Corporation

20630 Nordhoff Street, Chatsworth, CA 91311. Telephone: (213) 998-1800. TWX: (910) 494-2093.

Built and backed by Pertec Computer Corporation, (world's leading independent producer of computer peripheral equipment and distributed processing and data entry systems).

This 8-bit machine, by itself, is as versatile as a lot of systems that include peripherals

Skeptical? For starters, because of its unique design the H8 is the only machine in its price class that offers full system integration, yet, with just 4K of memory and using only its "intelligent" front panel for I/O, may be operated completely without peripherals!

Memory Display

Register Display

I/O Port Display

In addition, by using the features of its built-in Pam-8 ROM panel control program, the H8 actually allows you to dig in and examine machine level circuitry.

Responding to simple instructions the "intelligent" panel displays memory and register contents, lets you inspect and alter them even during operation. And for greater understanding, the front panel permits you to execute programs a single instruction at a time. The result is a powerful, flexible learning tool that actually lets you "see" and confirm each detail of H8's inner workings.

If you need further evidence, consider the fact that H8's system orientation allows you an almost unlimited opportunity for growth.

Memory is fully expandable, the 8080A CPU extremely versatile, and with the addition of high speed serial and parallel interfacing you gain the added flexibility of I/O operation with tape, CRT consoles, paper tape reader/punches, and soon floppy disk systems!

The H8 offers superior documentation including complete step-by-step assembly and operation manuals, is backed by 54 years of Heath reliability, and comes complete with BASIC, assembler, editor, and debug software — others charge over $60 for!

H8, simplicity for the beginner, sophistication for the expert and at $375* just right for you.

*Prices are mail order net FOB, Benton Harbor, Michigan. Prices and specifications subject to change without notice.

Be sure to use coupon on page 28 of this magazine to order your FREE Heathkit Catalog!

HEATHKIT COMPUTERS

System Engineered for Personal Computing

H8

CIRCLE INQUIRY NO. 26